

# Applications of Neural Networks and Fuzzy Logic to Integrated Water Management

## Project Report



Editors

A.H. Lobbrecht

Y.B. Dibike

D.P. Solomatine

Delft  
October, 2002



## ***Preface***

This report is produced as a result of a joint project of the Foundation for Applied Water Research (STOWA) and Delft Cluster research programme. IHE-Delft was responsible for the implementation of the project during a period of 2 years.

The supervisory committee for the project consisted of the following persons:  
Z.C. Vonk (chairman), B. van der Wal, J. van Dansik, P. van der Veer, C.J.H. Griffioen and P. Salverda.

The project team was formed of the two IHE staff members, A.H. Lobbrecht and D.P. Solomatine, and a group of the IHE research fellows – Y.B. Dibike, B. Bazartseren, B. Bhattacharya and L. Wang.

Important counterparts for performing the case studies were P. Vergouwe and S.-P. Bakker.

We would like to express our thanks to the supervisory committee for its thorough guidance, assistance and support.

Project- related materials can be found on Internet: <http://www.stowa-nn.ihe.nl>. Presentations made at the Symposium that was organized in the framework of the project and additional research materials can be found at <http://datamining.ihe.nl>.

Arnold H. Lobbrecht  
Yonas B. Dibike  
Dimitri P. Solomatine

editors



## General Contents

Project Summary	i
PART I: ARTIFICIAL NEURAL NETWORKS AND FUZZY LOGIC FOR INTEGRATED WATER MANAGEMENT: REVIEW OF THEORY AND APPLICATIONS	
Chapter 1 Introduction	5
Chapter 2 Neural Networks and their Applications	13
Chapter 3 Fuzzy Logic Approach and Applications	53
Chapter 4 Neuro-Fuzzy and Hybrid Approaches	67
Chapter 5 Discussion, Conclusions and Recommendations	75
PART II: ARTIFICIAL NEURAL NETWORKS FOR RECONSTRUCTION OF MISSING DATA AND RUNOFF FORECASTING: APPLICATION TO CATCHMENTS IN SALLAND	
1 Introduction	97
2 Data Preparation	99
3 Artificial Neural Networks	103
4 Application of Artificial Neural Networks	107
5 Results and Discussions	111
6 Conclusion and Recommendations	125
7 Reference:	127
PART III: ARTIFICIAL NEURAL NETWORKS AND FUZZY LOGIC SYSTEMS FOR MODEL BASED CONTROL: APPLICATION TO THE WATER SYSTEM OF OVERWAARD	
1 Introduction	133
2 Simulation and Control of Water Systems	137
3 Data Analysis	143
4 Aquarius Model for Overwaard	149
5 Artificial Neural Networks and Fuzzy Adaptive Systems	159
6 Application of ANN and FAS for Optimal Control	165
7 Conclusion and Recommendations	171
8. References	173



# Applications of Neural Networks and Fuzzy Logic to Integrated Water Management

## Project Summary

### Introduction

Management and control of water resources is a complex multi-disciplinary task requiring the adequate approaches and techniques. During the last decade considerable changes have been observed in approaches to tackling the problems of management and control. Most important were:

- introduction of the advanced information technology - personal computers and software, GPS systems, telecommunication networks. In water management this allowed for large-scale data collection campaigns, building data banks with the water-related data, increased level of automation of various tasks in control, etc.
- quantum leap in the amount of computer-based modelling. Modelling systems became an important part of the instrumentarium of engineers and managers providing the possibilities for *model-based control*.
- shift to more economical, optimal solutions. The increased competitiveness of various areas of human activities and political pressures lead to seeking optimal managerial and control actions where previously simply actions that are "good enough" would do. Flood management decisions for example, should follow the multi-objective approach, balancing various interests in minimizing damage.

All the mentioned shifts inevitably change the way water resources are managed and controlled, giving rise of attention to the so-called *hydroinformatics systems*. Such systems incorporate the latest advances in telecommunications, computing, computer-based modelling, artificial (computational) intelligence, machine learning, data analysis and processing, optimization and the associated decision support systems (DSS).

Traditional modelling of physical processes is often named *physically-based modelling* because it tries to explain the underlying processes (eg., hydrodynamic models based on Navier-Stokes partial differential equations numerically solved using finite-difference scheme). On the contrary, the so-called *data-driven models*, borrowing heavily from *artificial intelligence* (machine learning) techniques, are based on a limited knowledge of the modelling process and rely on the data describing input and output characteristics. Data-driven modelling uses results from such overlapping fields as data mining, artificial neural networks (ANN), rule-based type approaches such as expert systems, fuzzy logic concepts, rule-induction and machine learning systems. Sometimes "hybrid models" are built combining both types of models.

In this project, applications of two mostly widely used particular types of data-driven models, namely artificial neural networks (ANN) and fuzzy logic-based models, to modelling in the water resources management field are considered.

Neural network and fuzzy logic have been successfully applied to a wide range of problems covering a variety of sectors. Their practical applications, especially of neural networks expanded enormously starting from mid 80s till 90s partly due to a spectacular increase in computing power. During the last decade ANN evolved from being only a research tool into a tool that is applied to many real world problems: physical system control, various engineering problems, statistics, medical and biological fields. Consequently they are applied more and more in water management field as well.

There is a number of other methods attributed to artificial intelligence (machine learning): decision and model trees, Bayesian methods etc. Whatever models used, they are just techniques, methods of analysis and prediction that assist decision makers in making decisions. Models enhance these decisions only if used by experts in a proper way.

## **Objectives of the project**

The objectives of this project were:

- to review the principles of various types and architectures of neural network and fuzzy adaptive systems and their applications to integrated water resources management. Final goal of the review was as exposing and formulating progressive direction of their applicability and further research of the AI-related and data-driven techniques application in the water resources management field.
- to demonstrate applicability of the neural networks, fuzzy systems and other machine learning techniques in the practical issues of the regional water management. Two case studies were selected for that: Hoogheemraadschap van de Alblasserwaard en de Vijfheerenlanden (particularly, watersystem of Overwaard) and the Waterschap Groot Salland.

## **Main results and conclusions**

### *Review of applications*

Total of 85 papers, 14 theses and 15 books were reviewed. The published sources and the experience of the authors allow to formulate the advantages and recommendations of using ANN and fuzzy logic concepts for water related problems as follows:

- they give a possibility to complement or even to replace traditional (physically-based) methods
- the domain specific knowledge is required to a lesser accuracy than that for building physically based models
- data-driven models are much faster than physically-based models based on numerical solutions of partial differential equations
- application of data-driven methods require proper preparation of modelling exercises - analysis of logical relations between dependent and independent variables, choice of these variables, non-stochastic character of these relations, proper data collection and pre-processing, etc. They are considered in the report.
- methods like ANN and FRBS and many other methods of artificial intelligence, machine learning and data mining are in fact mathematical and modelling apparatus that have a general nature and can be applied practically in any area (as, for example differential equations). The success of their application depend mainly on the amount of available relevant data and on the experience of a modeller, leaving a lot to the "art of modelling"



- in the area of process modelling and/or simulation, these techniques were found useful for approximating the conventional models for saving computational power and for identification and learning the relationships and patterns on the basis of measured data for processes which are too complex to be described by physically-based models.

*ANNs* are extensively applied for assessment purposes like rainfall-runoff modelling, water quality prediction in natural flows, approximating ecological relations. They have also been applied for optimal reservoir operation. A remarkable number of publications on application of fuzzy logic approach for process control in wastewater treatment plants for deriving optimal control actions are available. Problem of real-time optimal operation of water related systems has been investigated by using neural networks, fuzzy logic approach and with neuro-fuzzy approach.

*Fuzzy-based methods* are applied successfully for identifying optimal control actions of wastewater treatment plant, determining optimal dosage thereof and determining leakage. They often are used in combination with the expert knowledge. *Fuzzy rule-based systems (FRBS)* (capable of building rules automatically) have been applied for drought prediction, determining optimal control action of polder pumping station and filling in gaps in the measured data. They have proven its ability to learn as good as *ANNs*.

*Neuro-fuzzy systems* has been applied successfully for detecting and identifying faults due to any measurement error, leakage or wrong valve status in water distribution system.

### *Case studies*

#### *Case study 1.*

#### *Artificial Neural Networks for Reconstruction of Missing Data and Runoff Forecasting: Application to Catchments in Waterschap Groot Salland*

Several time-series data on precipitation, evaporation, surface water level were available for this catchment. Preliminary data analysis showed that the hydrological time series has significant number of missing values and inconsistencies. The final application was focussed on two drainage areas (Rietberg and Stuw 7A) and time periods with reasonably consistent data. The outflow weir at Rietberg drains an area of 6,646 ha while stuw 7A drain areas 13,697 ha. Salland is generally a gently sloping area where water management is carried out with the help of fixed weirs, controlled weirs and irrigation pumping units operated by the water board of Groot Salland.

Two methods of using ANN, namely global neural network (GANN) and local neural networks (LANN) were considered. GANN considers all available time series data in its entirety while to be able to build LANN models, the complete time-series data has to be split into more homogeneous sub-sets so that the highly non-linear behaviour of the entire runoff process is captured in different classes for which the input-output relationships can be relatively simple. In general, LANNs have outperformed the GANNs for both problems of filling missing data and runoff forecasting. Moreover, using short-term history of water system variables as inputs to the network gave the best results. Once the ANN models are built, they are used to estimate values for missing runoff data and forecast a one-day ahead

discharge on the basis of available meteorological data such as measured rainfall, evaporation and discharge.

It must be mentioned that the operation of weirs and pumping stations in the area affects very much the homogeneity of input-output relationships in the data sets. As a result, ANN prediction of runoff values may not always match the monitored values. This could be due to the fact that manually operated weirs and discharge outlet structures control the flow in the drainage area and interfere with the natural flow and this, in its turn, affects the predictability of system behavior. However, the success of these ANN models in replicating the systems behaviour could be further improved by including information about the operational data of those regulating structures. The results could also be improved by classifying the data, not only by seasonal variations, but also by the magnitude of runoff events in the database. Moreover, it is important to frequently update the models by additional training or complete retraining every time new data set is available so that the models reflect the latest state of the system being modelled.

In general, the case studies on the catchments in Salland clearly demonstrated the applicability of artificial neural networks for runoff forecasting and filling of missing data in hydrological time series based on meteorological and other hydrological data.

#### *Case study 2.*

*Artificial Neural Networks and Fuzzy Logic Systems for Model Based Control: Application to the Water System of Overwaard (Hoogheemraadschap Alblasserwaard en de Vijfheerenlanden)*

Overwaard is a drainage basin located in South-Holland. The water system at Overwaard comprises of 22 drainage areas covering a total surface area of approximately 15,000 ha. First, a physically based distributed model of the water system was built (with the modelling system AQUARIUS) and calibrated with measured water levels and discharge data. The AQUARIUS model was found to be very effective in simulating the water system of Overwaard. Calibration results were acceptable since the simulated water level and discharge values were very much comparable to the observed ones. It has also been demonstrated with this model that central dynamic control can perform better than local control in cases of extreme precipitation events. Therefore, ANN and FAS were trained with the data generated by AQUARIUS model (run under central dynamic control mode) to replicate the central dynamic control's optimal pumping strategy for the main pumping station. External controllers were then designed using the trained ANN and FAS.

Online implementation of the trained ANN and FAS as external controllers was very successful and they were able to reproduce the centralised behaviour (in terms of water levels and corresponding discharges) of optimal control action by using easily measurable local information. The main advantage of the external intelligent controller is that it needed only one tenth of the simulation time of the one required by the central optimal controller of AQUARIUS. Replacing the slow computational component by the fast-running intelligent controllers in the way described in this study is believed to enhance the use of AQUARIUS in real time control tasks.

In general, the study clearly demonstrated the applicability of artificial neural network and fuzzy logic technologies for water management and control by considering the water system of Overwaard as an example.

### *Conclusion*

Overall, the objectives of the project have been reached: applicability of the neural networks, fuzzy systems and other machine learning techniques in the practical issues of the regional water management has been demonstrated. It can be also concluded that the cooperation between STOWA and the project “Data mining, knowledge discovery and data-driven modelling” of the Delft Cluster was beneficial to both parties: it allowed to combine the technologies developed and tested in the Delft Cluster project and to apply them to complex problems of water management and modelling that are encountered by the waterboards.

### *Recommendations for the future*

One of the recommendations is to make an inventory of other data-driven and machine learning techniques e.g. induction trees, advanced cluster analysis methods, non-linear dynamics (chaos theory), wavelet analysis, statistical learning theory (support vector machines) which has already proved to be effective data analysis and modelling methods.

Another potentially efficient approach is the so-called reinforcement learning. It is especially applicable in the problems of control. Our experience shows that the accuracy of a data-driven model used in water control can be significantly improved when different approaches are combined, e.g. ANN being complemented by reinforcement learning techniques.

In spite of the multiple successful experiments and applications described in the literature, it can be stated that acceptance of artificial neural networks (ANN) and fuzzy rule-based systems (FRBS) in water-related industries is slower than in other industries (chemical processing, electrical engineering, electronics, oil and gas exploration, military etc.). Still much to be done in “bringing the message” to the practitioners through refined research into less explored areas of data-driven modelling and machine learning, demonstrations of convincing experiments and promising prototype applications in various areas of water management.

Based on the successful applications of ANN, Fuzzy systems and other machine learning methods in this project it is therefore recommended to continue research in the area of data-driven and machine learning techniques, with applications to the problems of regional water systems management.





# **Artificial Neural Networks and Fuzzy Logic for Integrated Water Management: Review of Theory and Applications**

**Project report**

**Part I**

By

B. Bazartseren  
B. Bhattacharya  
A.H. Lobbrecht  
D.P. Solomatine

**Delft  
October, 2000**



## Contents of Part 1

<b>CHAPTER 1 INTRODUCTION.....</b>	<b>5</b>
1.1 GENERAL .....	5
1.2 PHYSICALLY-BASED AND DATA-DRIVEN MODELS .....	6
1.3 OBJECTIVE OF THE STUDY.....	8
1.4 ANN AND FUZZY LOGIC TECHNIQUES FOR WATER MANAGEMENT.....	8
1.5 OUTLINE .....	11
<b>CHAPTER 2 NEURAL NETWORKS AND THEIR APPLICATIONS.....</b>	<b>13</b>
2.1 INTRODUCTION.....	13
2.2 BASIC ELEMENTS IN NEURAL NETWORK STRUCTURE .....	13
2.3 NETWORK TOPOLOGY AND LEARNING ALGORITHMS.....	14
2.3.1 <i>Neural network structures</i> .....	14
2.3.2 <i>Error backpropagation networks</i> .....	15
2.3.3 <i>Radial-basis function networks</i> .....	17
2.3.4 <i>Recurrent neural networks</i> .....	19
2.3.5 <i>Self-Organising feature maps and other cluster analysis techniques</i> .....	23
2.3.6 <i>Principal component NN</i> .....	27
2.4 APPLICATIONS OF NEURAL NETWORKS IN THE WATER SECTOR.....	29
2.4.1 <i>Drinking water systems</i> .....	29
2.4.2 <i>Sewerage systems</i> .....	31
2.4.3 <i>Inland water systems</i> .....	37
2.4.4 <i>Coastal water systems</i> .....	45
2.5 PRACTICAL ISSUES OF USING NN FOR ENGINEERING APPLICATIONS .....	48
2.5.1 <i>Introduction</i> .....	48
2.5.2 <i>Analysing the problem</i> .....	48
2.5.3 <i>Data preparation and analysis</i> .....	48
2.5.4 <i>Model selection and building</i> .....	49
2.5.5 <i>Training and testing the network</i> .....	50
2.5.6 <i>Output and error analysis</i> .....	51
2.5.7 <i>Implementation of a neural network based project</i> .....	51
<b>CHAPTER 3 FUZZY LOGIC APPROACH AND APPLICATIONS.....</b>	<b>53</b>
3.1 INTRODUCTION.....	53
3.2 BASIC CONCEPT OF FUZZY LOGIC APPROACH.....	53
3.3 FUZZY ADAPTIVE SYSTEMS .....	56
3.4 FUZZY LOGIC CONTROL.....	58
3.5 APPLICATION OF FUZZY LOGIC APPROACHES.....	59
3.5.1 <i>Sewerage systems</i> .....	59
3.5.2 <i>Inland water systems</i> .....	63
<b>CHAPTER 4 NEURO-FUZZY AND HYBRID APPROACHES.....</b>	<b>67</b>
4.1 INTRODUCTION.....	67
4.2 NEURO-FUZZY HYBRID SYSTEM.....	67
4.3 NEURO-FUZZY ARCHITECTURE .....	68
4.4 OTHER HYBRID APPROACHES.....	70
4.5 APPLICATION OF NEURO-FUZZY SYSTEMS.....	71
4.5.1 <i>Drinking water systems</i> .....	71
4.5.2 <i>Sewerage systems</i> .....	72
4.4.3 <i>Inland water systems</i> .....	73

**CHAPTER 5 DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS..... 75**  
5.1 DISCUSSION ..... 75  
5.2 CONCLUSIONS ..... 80  
REFERENCES ..... 83  
APPENDIX ..... 88



# Chapter 1 Introduction

## 1.1 General

Management and control of water resources is a complex multi-disciplinary task requiring the adequate approaches and techniques. During the last decade considerable changes have been observed in approaches to tackling the problems of management and control. We will mention only three of them.

1. Introduction of the advanced information and communication technology (ICT) devices - personal computers, GPS systems, telecommunication networks and associated processors. In water management the power of these devices and the associated software allowed for large-scale data collection campaigns, building data banks with the water-related data, increased level of automation of various tasks in control, etc.
2. Quantum leap in the amount of computer-based modelling. Modelling systems became an important part of the instrumentarium of engineers and managers providing the possibilities for *model-based control*. Important decisions in water management are now impossible without the enhanced systems and scenario analysis based on modelling various alternatives. Models of surface and ground water flows have become more accurate due to the amount of the refined modelling techniques, the availability of data for their calibration and the computing power allowing for more accurate schematization, finer grid etc.
3. Shift to more economical, optimal solutions. The increased competitiveness of various areas of human activities and political pressures lead to seeking optimal managerial and control actions where previously simply actions that are "good enough" would do. Flood management decisions for example, should follow the multi-objective approach, balancing various interests in minimizing damage.

All the mentioned shifts inevitably change the way water resources are managed and controlled, giving rise of attention to the so-called *hydroinformatics systems* (Fig. 1). Such systems incorporate the latest advances in telecommunications, computing, computer-based modelling, artificial intelligence, data analysis and processing, optimization and the associated decision support systems (DSS). Several examples of hydroinformatics applications for flood warning and risk assessment projects could be mentioned (eg., resulting from EU projects TELEFLEUR and EUROTAS with Dutch participation).

In case of TELEFLEUR design, such system receives signals from rain gauges through communication lines, and data from meteorological models, this data is fed into the hydrological models and data-driven predictive models which produce predictions of water levels, this information is combined with the facts from knowledge-based systems and given to the decision makers. A similar system could be foreseen in the context of water management in polder areas.

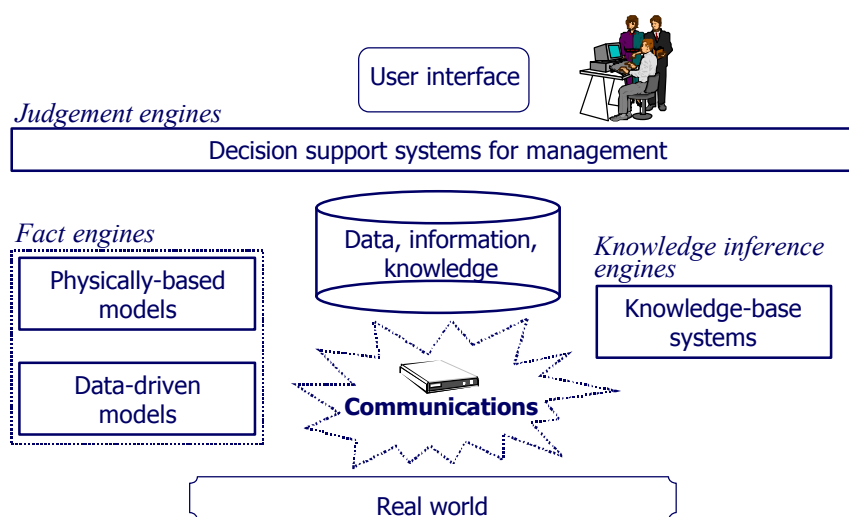


Figure 1.1: Typical hydroinformatics system

## 1.2 Physically-based and data-driven models

Traditional modelling of physical processes is often named *physically-based modelling* (or knowledge-driven modelling) because it tries to explain the underlying processes. An example of such a model is a hydrodynamic model based on Navier-Stokes partial differential equations numerically solved using finite-difference scheme.

On the contrary, the so-called *data-driven models*, borrowing heavily from Artificial Intelligence (AI) techniques, are based on a limited knowledge of the modelling process and rely on the data describing input and output characteristics. These methods, however, are able to make abstractions and generalizations of the process and play often a complementary role to physically-based models. Data-driven modelling uses results from such overlapping fields as data mining, artificial neural networks (ANN), rule-based type approaches such as expert systems, fuzzy logic concepts, rule-induction and machine learning systems. Sometimes "hybrid models" are built combining both types of models.

A simple example of a data-driven model is a linear regression model. Coefficients of the regression equation are identified ("trained") on the basis of the available existing data. Then for a given new value of the independent (input) variable it gives an approximation of an output variable value. More complex data-driven models are highly non-linear, allowing many inputs and many outputs (Figure 1.2) They need a considerable amount of historical data to be trained, and if this is done properly, they are able not only to approximate practically any given function, but also to generalise, providing correct output for the previously "unseen" inputs.

Apart from function approximation and regression data-driven techniques are widely used in solving *classification* problems, that is grouping data into classes. *Unsupervised* learning methods often incorporate self-organizing features, enabling them to find unknown regularities, meaningful categorization and patterns in the presented input data. *Supervised* learning allows to train classifiers able to attribute new data to known classes.

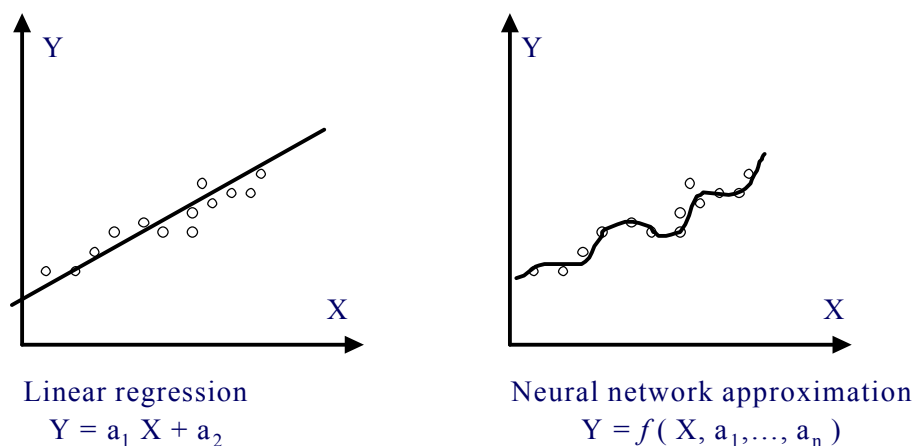


Figure 1.2: Data-driven models: linear regression and ANN. Data-driven models are based on pure relationships between input ( $X$ ) and output ( $Y$ ) data and not the physical principle linking  $X$  and  $Y$ .

For a regression equation, coefficients  $a_1$  and  $a_2$  have to be identified (trained) by solving optimization problem on the basis of the available data. For ANN, many more coefficients have to be trained but it can reproduce non-linear multi-dimensional relationships.

Scientific and engineering community has acquired already an extensive experience in developing and using data-driven techniques (details on the experience of IHE-Delft, can be found on Internet at [www.ihe.nl/hi/sol](http://www.ihe.nl/hi/sol)). Not all sectors of water industry, however, have used advantages of these methods.

In this review, applications of two mostly widely used particular types of data-driven models, namely artificial neural networks (ANN) and fuzzy logic-based models, to modelling in the water resources management field are considered.

*Artificial neural network (ANN)* is an information processing system that roughly replicates the behaviour of a human brain by emulating the operations and connectivity of biological neurons. From a mathematical point of view ANN is a complex non-linear function with many parameters that are adjusted (calibrated, or trained) in such a way that the ANN output becomes similar to the measured output on a known data set.

The origin of *fuzzy logic* approach dates back to 1965 since Lotfi Zadeh's introduction of fuzzy-set theory and its applications. Since that period fuzzy logic concept has found a very wide range of applications especially in the industrial systems control that are very complex, uncertain and cannot be modelled precisely, even under various assumptions and approximations. An example of a fuzzy rule is:

IF precipitation = *high* AND Reservoir-level = *medium*  
THEN water-release = *medium*

(here precipitation, reservoir-level and water-release are so-called linguistic variable with fuzzy values *medium*, *high* etc.). In this review two main types of *fuzzy rule-based systems (FRBS)* are considered: (a) fuzzy inference systems, which work on already constructed rule-base mainly on the basis of expert knowledge, and (b) fuzzy adaptive systems, which can also build and adjust rule-base automatically on the basis of a given training set.

Neural network and fuzzy logic have been successfully applied to a wide range of problems covering a variety of sectors. Their practical applications, especially of neural networks expanded enormously starting from mid 80s till 90s partly due to a spectacular increase in computing power (Kappen, 1996). During the last decade ANN evolved from being only a research tool into a tool that is applied to many real world problems: physical system control, various engineering problems, statistics, medical and biological fields. Consequently they are applied more and more in water management field as well.

It should be noted that water resources management is a complex issue having a wide range of activities. It is *an application of structural and nonstructural measures to control natural and man-made water resources systems for beneficial human and environmental purposes* (Crigg, 1996). It becomes much more complex than any other management problem due to interdependence of several sectors of water resources. In order to have a systematic review, the application of ANN and fuzzy logic approach to water resources management problem has been classified into several distinctive activities and application sectors.

### 1.3 Objective of the study

The objective of this review relates to understanding the principles of various types and architectures of neural network and fuzzy adaptive systems and reviewing their applications for integrated water resources management. Final goal of the review can be described as exposing and formulating progressive direction of further research of the data-driven and AI techniques application in the water resources management field.

### 1.4 ANN and Fuzzy logic techniques for water management

A wide range of application of ANN and Fuzzy logic techniques has been investigated in the field of water resources management. As mentioned before, the water resources management is a highly complex issue covering a wide spectrum of activities in the field of assessment, planning, designing, operation and maintenance (Figure 1.3). As in any other management field, all the above activities take place in institutional, social and political environment, which is not intended to emphasize in this report. From more general point of view, AI techniques can be applied for prediction, simulation, identification, classification and optimization. For water resources management field those can be described as follows:

*Simulation (physically-based) models.* Deterministic models are used for simulation of various processes related to the management of water such as hydrodynamic, morphological, ecological, water quality, groundwater flow etc. All these models use detailed description and fine quantization of the undergoing processes. On the contrary, neural networks do not require the explicit knowledge of physical processes and the relations can be fitted on the basis of measured data. At the same time, the neural networks or fuzzy adaptive systems can approximate any logical condition action pairs with reasonable accuracy. In many or most occasions it was shown that the neural networks tend to give better result than the deterministic models, provided that the process under consideration is not changed in time.

*Prediction.* If significant variables are known, without knowing the exact relationships, ANN is suitable to perform a kind of function fitting by using multiple parameters on the existing information and predict the possible relationships in the coming future. This sort of problem includes rainfall-runoff prediction, water level and discharge relations, drinking water

demand, flow and sediment transport, water quality prediction etc. Also filling or restoring of missing data in a time series can be considered as a kind of prediction.

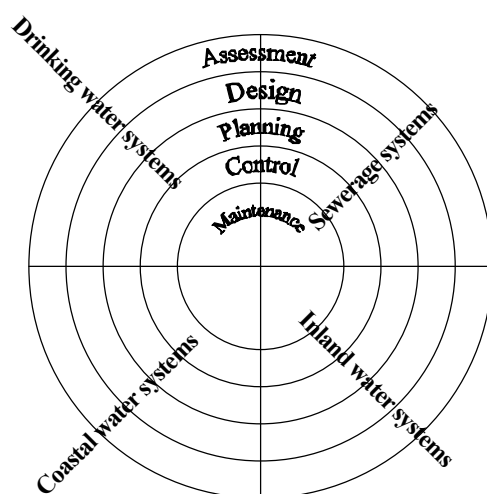


Figure 1.3: Schematization of different activities and sectors in water management

*Identification and classification.* In order to represent data more efficiently, it is needed to extract the most important features in the data set. The final goal of feature extraction in fact is a classification. Unsupervised neural networks often incorporate self-organizing features, enabling them to find unknown regularities, meaningful categorisation and patterns in the presented input data.

*Optimization.* The common task of making decisions in water resources management problem normally includes multiple objectives to be optimised taking into account many different constraints. Neural networks or fuzzy logic approaches are not optimization techniques. However, by making use of their generalization ability they approximate either the optimal solution or optimise through continuously training their weights (neural networks) or their membership functions (fuzzy logic approach).

As mentioned earlier there are 5 main activities in water resources management and each of them can have its subactivities. The activities can be described briefly as follows:

### 1. Assessment

- a. *Resources or quantity assessment* In this sub-activity the quantitative aspects such as estimation of resources in surface and groundwater system are included. For example, rainfall-runoff modelling is one of the areas where neural network is mostly applied. Modelling the physics of process such as of forming streamflow from rainfall in the area may not always be feasible. The reasons for that might be most of the quantitative processes are complex and dynamic. The processes vary in time and space and lack necessary data for modelling. On the other hand, if it is modelled precisely a lot of effort is required for model calibration, which makes AI applicable.
- b. *Ecological relations* Ecological models use a mathematical description of physical and chemical processes, which are very complex and non-linear in nature. Usually the relationships of ecological variables are derived empirically and most of the time they are linear approximation of the processes, where all the influencing effects may not be

considered. Although the results of deterministic modelling are adequately good, in case of modelling measurable ecological variables, the neural networks are found to be better to generalize the complex relationships.

- c. *Water quality management* Water quality management problem is mostly based on imprecise and insufficient information. Most of the time, goals or constraints may not be defined precisely due to the fact that they are based on ill-defined and subjective requirements of human judgement or preferences. Although, the numerical models are available for water quality simulation, the uncertainties and imprecision are not well covered in those models. Furthermore, the need for calibration of water quality models makes the neural networks advantageous over these models. Range of this type of problem varies from water quality of subcatchment surface water to water quality of the urban drainage and drinking water supply systems.

## 2. *Designing*

This activity includes the analysis and design of engineering structures for water resources management. Structures for water management can be classified into several classes according to their purpose or function: water supply, wastewater, storm water, hydropower, navigation and environmental protection. The designing of these structures should not be considered as a modelling problem where difficulties are encountered in describing it mathematically. Engineers design the structures on the basis of given conditional data. However, the simple structure design might be learnt by AI techniques. Hitherto no application of AI techniques for structure designing is published.

## 3. *Planning*

Planning activities considered in this class are operational planning such as water demand prediction, reservoir operation etc. In other words, the problems of operational planning have been classified in this category. As an example, analysing the influencing parameters for operational planning and consequently predicting the future action is one of the important issues for planning and management for water authorities. The performance of statistical prediction models is not satisfactory in many cases. Use of AI techniques possibly makes it more reliable for these kinds of problems where traditional techniques are not very successful. At the same time, AI techniques can be used to replicate the optimal operation planning from optimization problem or can be used in optimization loop.

## 4. *Operation*

These activities include the operation and real-time control of water systems. The relation between the optimal decision or action and the influencing parameters can be learned by neural networks. Also it is possible to use these relations for deriving the decision and control actions in real-time. The regional or subcatchment water resources system management and control, urban water management problems such as water and wastewater treatment and drinking water supply can be included in this field of activities.

## 5. *Maintenance*

A common example of maintenance problem is fault detection in the water system, such as distribution system or treatment plant system. The faults are very uncertain in nature and create difficulty in distinguishing the cause of fault. There can be many

different criteria to cause faulty operation such as leakage, wrong valve status and measurement error due to the telemetry system failure. By using AI techniques it is possible to identify the possible cause of failure in the system.

In order to classify the applications of ANN and Fuzzy Adaptive Systems systematically, we distinguish the following application sectors (figure 1.1) in this review:

- drinking water systems (quality and quantity in piped community water distribution)
- sewerage systems (storm water collection systems, drinking water purification plants, sewer water treatment plants)
- inland water systems (quality and quantity issues in surface and groundwater resources systems including engineering structures such as reservoir, dams, irrigation systems etc)
- coastal water systems (quantity aspects in coastal water management problems, navigation and related engineering structure problems)

## 1.5 Outline

The overview is organised in five chapters.

Chapter 2 introduces the basic understanding of various neural network topology and learning algorithms and their application in specific application sectors of integrated water resources management such as drinking water systems, sewerage systems, inland water systems and coastal water systems. This chapter also includes some practical hints for working successfully in neural network based projects.

Chapter 3 gives the basic introduction to the general fuzzy logic approach and Fuzzy Adaptive Systems with function approximation and learning capability. Moreover, the application of the techniques in the water management field is reviewed.

Chapter 4 introduces the neuro-fuzzy approach, which takes advantages of neural network as well as fuzzy logic approaches. Most of the literature reveals that the approach is becoming an interesting field of artificial intelligence research. The chapter also includes the overview of the application in the related area.

Chapter 5 gives a conclusion and recommendation for possible research.





## Chapter 2 Neural networks and their applications

### 2.1 Introduction

One of the most popular data-driven techniques attributed by various authors to machine learning, data mining, soft computing etc. is an Artificial Neural Network (ANN). An ANN is an information processing system that roughly replicates the behaviour of a human brain by emulating the operations and connectivity of biological neurons (Tsoukalas and Uhrig, 1997). It performs a human-like reasoning, learns the attitude and stores the relationship of the processes on the basis of a representative data set that already exists. Therefore, generally speaking, the neural networks do not need much of a detailed description or formulation of the underlying process.

Depending on the structure of the network, usually a series of connecting neuron *weights* are adjusted in order to fit a series of inputs to another series of known outputs. When the weight of a particular neuron is updated it is said that the neuron is *learning*. The training is the process that neural network learns. Once the training is performed the verification is very fast. Since the connecting weights are not related to some physical identities, the approach is considered as a black-box model. The adaptability, reliability and robustness of an ANN depend upon the source, range, quantity and quality of the data set.

During the last decade ANNs evolved from only a research tool into a tool that is applied to many real world problems: physical system control, engineering problems, statistics, even medical and biological fields. The number of European patents obtained in the last decade corroborates the trend of increased applications of ANNs (Kappen, 1996). This chapter starts with a brief introduction of different structures and learning algorithms of neural networks. However, it is not aimed to cover the theory of each learning algorithm in detail. Applications of neural networks in the respective water resources management field are overviewed later in this section. At the end of the chapter, some practical hints on using neural network models are given, based on the handbooks written by experts.

### 2.2 Basic elements in neural network structure

As has been mentioned before, the ANN performs fundamentally like a human brain. The cell body in the human neuron receives incoming impulses via *dendrites* (receiver) by means of chemical processes (Figure 2.1). If the number of incoming impulses exceeds certain threshold value the neuron will discharge it off to other neurons through its *synapses*, which determines the impulse frequency to be fired off (Beale and Jackson, 1990).

Therefore, processing units or neurons of an ANN consists of three main components; synaptic weights connecting the nodes, the summation function within the node and the transfer function (see Figure 2.4). *Synaptic weights* characterise themselves with their strength (value) which corresponds to the importance of the information coming from each neuron. In other words, the information is encoded in these strength-weights. The *summation function* is used to calculate a total input signal by multiplying their synaptic weights and summing up all the products.

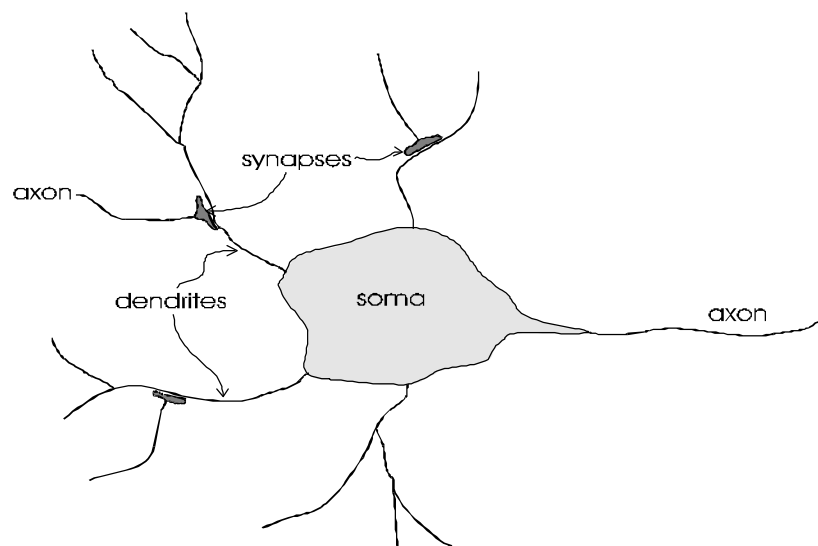


Figure 2.1: Schematisation of biological neuron

*Activation function* (or sometimes called a threshold function) transforms the summed up input signal, received from the summation function, into an output. The activation function can be either linear or non-linear. The type of activation function characterises the neural network. The most commonly used type of activation function is shown in Figure 2.5. An ANN consists of distinct layers of processing units and connecting weights.

## 2.3 Network topology and learning algorithms

### 2.3.1 Neural network structures

Structure of an ANN can be classified into 3 groups as per the by arrangement of neurons and the connection patterns of the layers: feedforward (error backpropagation networks), feedback (recurrent neural networks and adaptive resonance memories), self-organizing (Kohonen networks). Also neural networks can be roughly categorized into two types in terms of their learning features: *supervised* learning algorithms, where networks learn to fit known inputs to known outputs, and *unsupervised* learning algorithms, where no desired output to a set of input is defined. The classification is not unique and different research groups make different classifications. One of the possible classifications is shown in Figure 2.2.

The feedforward neural networks consist of three or more layers of nodes: one input layer, one output layer and one or more hidden layers. The input vector  $x$  passed to the network is directly passed to the node activation output of input layer without any computation. One or more hidden layers of nodes between input and output layer provide additional computations. Then the output layer generates the mapping output vector  $z$ . Each of the hidden and output layer has a set of connections, with a corresponding strength-weight, between itself and each node of preceding layer. Such structure of a network is called a *Multi-Layer Perceptron* (MLP). Figure 2.3 shows a typical multi-layer perceptron.

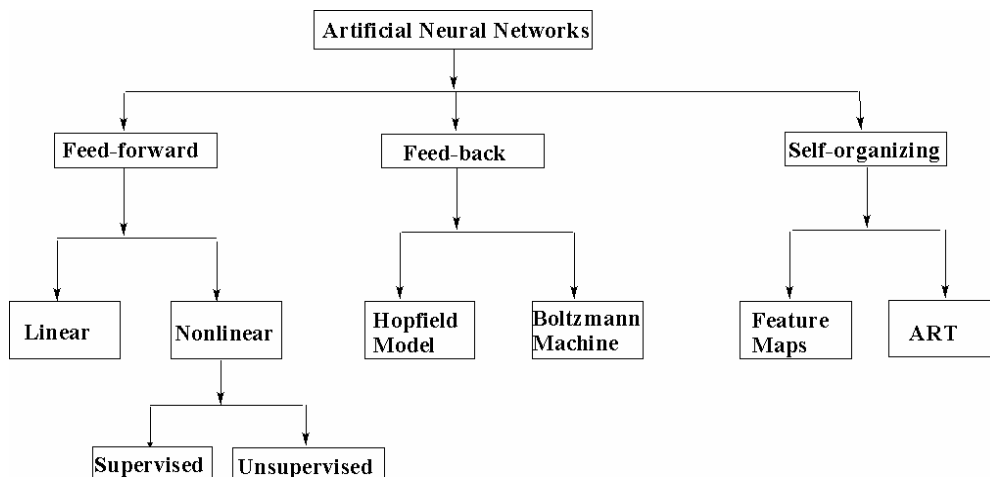


Figure 2.2: Neural network classification

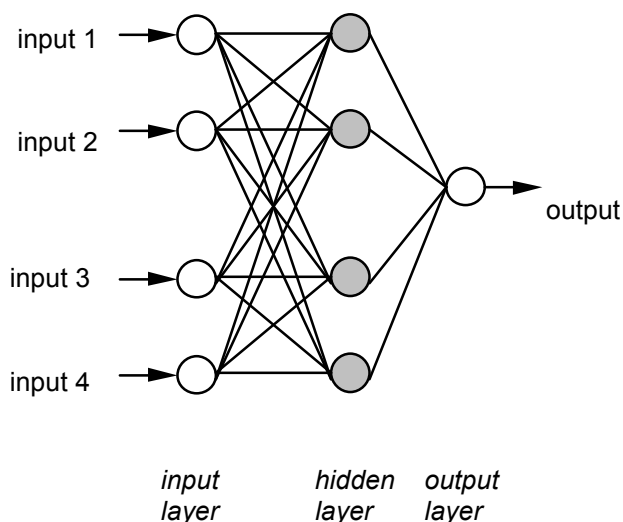


Figure 2.3: A fully connected multi-layer perceptron

The feedback neural networks have loops that feedback information in the hidden layers. In Self-Organising Feature Maps (SOFM) the multidimensional input space is mapped into two or three dimensional maps by preserving the necessary features to be extracted or classified. An SOFM consists of an input layer and an output map. Some of the commonly used feedforward and feedback neural networks are briefly discussed below.

### 2.3.2 Error backpropagation networks

The error backpropagation network (EBP) is one of the most commonly used types of neural networks. The EBP networks are widely used because of their robustness, which allows them to be applied in a wide range of tasks. The error backpropagation is the way of using known input-output pairs of a target function to find the coefficients that make a certain mapping function approximate the target function as closely as possible.

The task faced by a backpropagation neural network is that of learning supervised mapping: given a set of input vectors and associated target vectors, the objective is to learn a rule that

captures the underlying functional relationship between the input vectors and the target vectors. Mathematically, each target vector  $\vec{z}$  is a function,  $f$ , of the input vector  $\vec{x}$  :

$$\vec{z} = f(\vec{x}) \tag{2.1}$$

The task of the backpropagation network is to learn the function  $f$ . This is achieved by finding regularities in the input patterns that correspond to regularities in the output patterns. The network has a weight parameter vector, whose values are changed to modify a function  $f'$  computed by the network to be as close as possible to  $f$ .

The backpropagation network operates in two modes: mapping and learning. In mapping mode, each example is analysed one by one and the network estimates the outputs based on the values of the inputs. For every example, each input node passes a value of an independent variable  $x_i$  to all the nodes of the hidden layer. Each hidden node computes a weighted sum of the input values based on its weights  $a_i$  (Figure 2.4). The weights are determined during the learning mode. Finally, from this value of the weighted sum, the hidden nodes compute a sigmoid output  $y_i$  of the hidden nodes. The sigmoid function provides a bounded output of the hidden node. Each of the output nodes receives the outputs of the hidden nodes  $y_i$ , computes a weighted sum of the inputs based on the weights  $b_i$  and finally, determines the sigmoid output  $z_i$  of the node. The output of the output node,  $z_i$ , is the estimated value of the  $i^{\text{th}}$  dependent variable. The output from the output node is compared with the target output and the error is propagated back to adjust the connecting weights  $a$  as well as  $b$  and this procedure is called backpropagation.

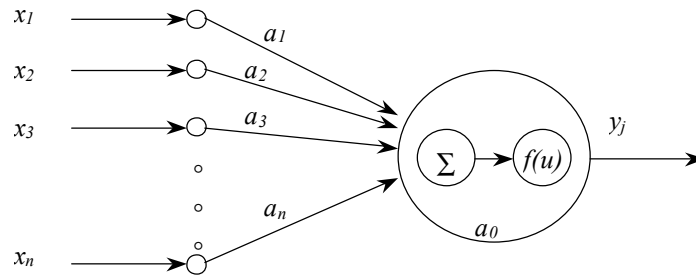


Figure 2.4: Computations within a single node

For an MLP, given the input vector  $X=(x_1, x_2, \dots, x_n)$ , the output from the hidden node will be as follows:

$$y_j = g(u) = g(a_{0j} + \sum_{i=1}^{N_{inp}} a_{ij}x_i) \tag{2.2}$$

Where  $j=1..N_{input}$  and  $a_{ij}$  is the weight of the  $i^{\text{th}}$  node for the  $j^{\text{th}}$  input. The outputs from the hidden nodes would be the input to the next hidden layer (if there is more than one hidden layer) or to the output nodes. The outputs of the output nodes should be calculated as follows:

$$z_k = g(b_{ok} + \sum_{j=1}^{N_{hid}} b_{jk}y_j) \tag{2.3}$$

Where  $k=1.. N_{output}$  and  $b_{jk}$  is the weight of the  $j^{\text{th}}$  node for the  $k^{\text{th}}$  output. The transfer function, mostly used a sigmoid or a logistic function (Figure 2.5), gives values in the range of  $[0,1]$  and can be described as:

$$g(u) = \frac{1}{1 + e^{-u}} \quad (2.4)$$

The mean square error is the way of measuring the fit of the data and is calculated as:

$$E = \frac{\sum_{n=1}^N \sum_{k=1}^K (z_{kn} - t_{kn})^2}{2NK} \quad (2.5)$$

where  $N$  is the number of examples in the data set,  $K$  is the number of outputs of the network,  $z_{kn}$  is the  $k^{\text{th}}$  actual output for the  $n^{\text{th}}$  example and  $t_{kn}$  is the  $k^{\text{th}}$  target output for the  $n^{\text{th}}$  example. For more details see Smith (1993).

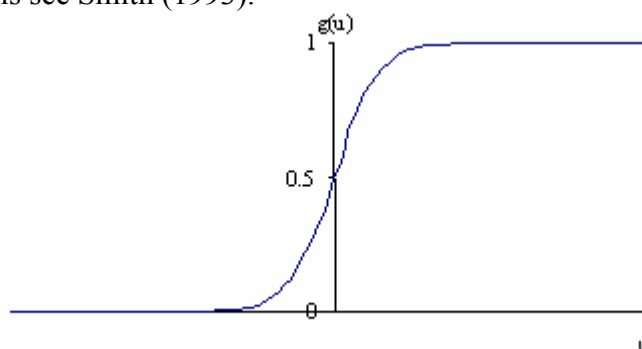


Figure 2.5: Sigmoid or logistic transfer function

In the learning mode, an optimization problem is solved to decrease the mean square error and it finds such a value for  $a$  and  $b$  to bring the  $E$  to minimum. By solving the optimization problem and knowing the slope of the error surface, the weights are adjusted after every iteration. As per the gradient descent rule the weights are adjusted as follows:

$$\Delta w(t) = -\mathbf{h} \frac{\partial E}{\partial w} + \mathbf{m} \Delta w(t-1) \quad (2.6)$$

where  $\mathbf{h}$  is the learning rate and  $\mathbf{m}$  is the momentum value.

### 2.3.3 Radial-basis function networks

A Radial Basis Function (RBF) is another type of feed-forward ANN. Typically in an RBF network, there are three layers: one input, one hidden and one output layer. Unlike the backpropagation networks, the number of hidden layer can not be more than one. The hidden layer uses Gaussian transfer function instead of the sigmoid function. In RBF networks, one major advantage is that if the number of input variables is not too high, then learning is much faster than other type of networks. However, the required number of the hidden units increases geometrically with the number of the input variables. It becomes practically impossible to use this network for a large number of input variables.

The hidden layer in RBF network consists of an array of nodes that contains a parameter vector called a 'radial centre' vector (Schalkoff, 1997). The hidden layer performs a fixed non-linear transformation with non-adjustable parameters. The approximation of the input-

output relation is derived by obtaining a suitable number of nodes in the hidden layer and by positioning them in the input space where the data is mostly clustered. At every iteration, the position of the radial centres, its width (variation) and the linear weights to each output node are modified. The learning is completed when each radial centre is brought up as close as possible to each discrete cluster centres formed from the input space and the error of the network's output is within the desired limit.

The centres and widths of the Gaussians are set by the unsupervised learning rules, and the supervised learning is applied to the output layer. For this reason RBF networks are called hybrid networks.

The learning algorithm is formulated as follows:

1. Find the centres for an RBF. In order to do that the following procedure is followed:
  - a. The number of the hidden nodes is chosen beforehand and the centres are assigned ( $w_j$ ) which are equally set to the randomly selected input vector  $x_j$  where in both cases  $j=1..J$ .
  - b. All the remainder of the training pattern is clustered into a class or cluster  $j$  of the closest centre  $w_j$  and the locations of each centre are calculated again using the Nearest Neighbour Rule.
  - c. The above steps are repeated until the locations of the centres stop changing.
2. The width  $\sigma$  of the radial centre for each hidden neuron is calculated. The distance between the centres of the clusters defines the width or variance.
3. Calculate the output from each hidden neuron as a function of a radial distance from the input vector to the radial centre. Calculated distance between the centre and the input vector is passed through a non-linear mapping function. Then the output can be written as  $y_i = f(d_j)$ . A distance measure, to determine how far is an input vector from the centre, usually is expressed as an Euclidean distance measure (Taylor, 1996). The distance  $\delta_j$  between the input vector  $X=(x_1, x_2, \dots, x_k)$  and the radial centre  $X_j=(w_{1j}, w_{2j}, \dots, w_{mj})$  is written as:

$$d_j = \sqrt{\sum_{i=1}^k (x_i - w_{ij})^2} \quad (2.7)$$

This mapping function on each hidden node is usually a Gaussian function of the following form:

$$f(d_j) = \exp(-1 d^2) \quad (2.8)$$

4. Weights ( $b_j$ ) for the output layer are calculated using methodologies such as the Least Square Method or the Gradient Descent Method. The output node then receives the values indicating how far is the example from each of them and combines the outputs linearly. The output from the output node can be described by the following equation:

$$z_k = \frac{\sum_{j=1}^J b_{jk} y_j}{\sum_{j=1}^J y_j} \quad (2.9)$$

- where  $b_{jk}$  – the weight on the connection from the hidden node  $j$  to the output node  $k$ ,  
 $y_j$  – the output from the hidden node  $j$
5. Calculate the error between the network's output and the target output and if the error of the network's output is more than the desired limit then the number of the hidden units are changed and all the steps are repeated again.

The advantage of this network is that the learning process can be faster than the backpropagation networks, although the accuracy of the solution is highly dependent on the range and quality of data (Dibike, 1997).

### 2.3.4 Recurrent neural networks

Recurrent neural networks (RNN) have a closed loop in the network topology. They are developed to deal with the time varying or time-lagged patterns and are usable for the problems where the dynamics of the considered process is complex and the measured data is noisy. Specific groups of the units get the feedback signals from the previous time steps and these units are called *context* unit (Schalkoff, 1997). The RNN can be either fully or partially connected. In a fully connected RNN all the hidden units are connected recurrently, whereas in a partially connected RNN the recurrent connections are omitted partially (see Figure 2.6). Examples of recurrent neural networks are Hopfield networks, Regressive networks, Jordan-Elman networks, and Brain-State-In-A-Box (BSB) networks.

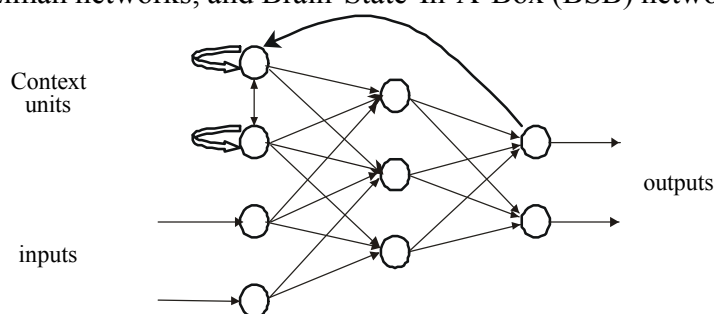


Figure 2.6: Example of partially connected recurrent neural network (Schalkoff, 1997)

All types of recurrent neural networks are normally trained with the backpropagation learning rule by minimizing the error by the gradient descent method. Mostly they use some computational units which are called associative memories or context units, that can learn associations among dissimilar binary objects, where a set of binary inputs is fed to a matrix of resistors, producing a set of binary outputs. The outputs are '1' if the sum of the inputs is above a given threshold, otherwise it is zero. The weights (which are binary) are updated by using very simple rules based on *Hebbian* learning. These are very simple devices with one layer of linear units that maps  $N$  inputs (a point in  $N$  dimensional space) onto  $M$  outputs (a point in  $M$  dimensional space). However, they remember the past events.

### Jordan-Elman networks

Jordan and Elman networks combine the past values of the context unit with the present input ( $x$ ) to obtain the present net output. The Jordan context unit acts as a so called *lowpass* filter, which creates an output that is the weighted (average) value of some of its most recent past outputs (see Figure 2.7). The output ( $y$ ) of the network is obtained by summing the past values multiplied by the scalar parameter  $t^n$ . The input to the context unit is copied from the

network layer, but the outputs of the context unit are incorporated in the net through their adaptive weights (see equation 2.10).

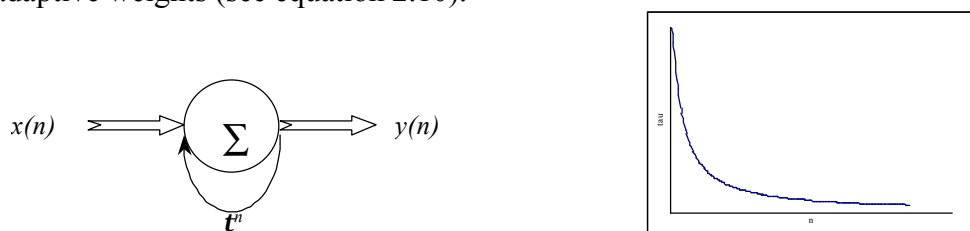


Figure 2.7: Parameters for Jordan-Elman network (*NeuroSolutions manual*)

$$y(n) = \sum_{i=0}^n x(n) t^{n-i} \tag{2.10}$$

In these networks, the weighting over time is inflexible since we can only control the time constant (i.e. the exponential decay). Moreover, a small change in time is reflected as a large change in the weighting (due to the exponential relationship between the time constant and the amplitude). In general, we do not know how large the memory depth should be, so this makes the choice of  $t$  problematic, without having a mechanism to adopt it.

In linear systems, the use of past input signals creates the moving average (MA) models. They can represent signals that have a spectrum with sharp valleys and broad peaks. The use of the past outputs creates what is known as the *autoregressive* (AR) models. These models can represent signals that have broad valleys and sharp spectral peaks. The Jordan net is a restricted case of a non-linear AR model, while the configuration with context units fed by the input layer is a restricted case of non-linear MA model. Elman’s net does not have a counterpart in linear system theory. These two topologies have different processing power (Beale and Jackson, 1991).

### Hopfield networks

Hopfield networks are the recurrent neural networks with no hidden units. The idea of this type of network is to get a convergence of weights to find the minimum value for *energy function*, just like a ball going down to the hill and stops when energy is converted to other form due to friction and other forces (Gurney, 1999). Also it can be compared to the vortices in a river. Taking input vector  $X$ , the system state and the network dynamics converge the energy function into a stable state or equilibrium point denoted as  $P$  (see Figure 2.8). After the network has learned and a new ball is presented on the top of the hill, it should remember where the ball has to stop.

Every node of the Hopfield net is connected to all other nodes but not to itself, so that the flow is not in a single direction. Even a node can be connected to itself in a way of receiving the information back through other nodes. Weights, the connection strengths are symmetric so that the weights from node  $i$  to node  $j$  are equal to the weight from node  $j$  to node  $i$ , which means  $w_{ij}=w_{ji}$  and  $w_{ii}=0$ .



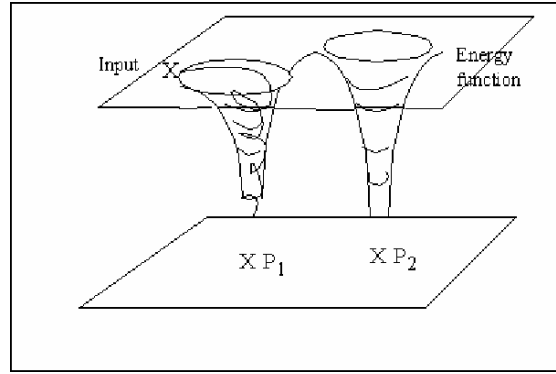


Figure 2.8: Simplified description of Hopfield network learning (NeuroSolutions manual)

The state of the network at given time is expressed by the vector of node outputs. At any given state the nodes are selected randomly and the output of the node is updated when the node is fired. The fired node evaluates its activation in a normal way and output of the node is '1' if it is greater or equal to zero and '0' otherwise. The network now finds itself exactly in the same state or in a new position, which is in a certain Hamming distance from the old one. In the next iteration, a node is chosen randomly which updates its weight and the system state. The procedure is repeated till the system reaches a stable state or minimum energy value, where no more update is desirable. The energy of the system for each pair of node is defined as follows:

$$e_{ij} = -w_{ij}x_i x_j \quad (2.11)$$

where  $x_i$  and  $x_j$  are node outputs and the  $w_{ij}$  is a weight. From equation (2.11) one can see the minimum system energy is achieved when both node outputs take a value of '1'. The total system energy is then found by summing all the energy for all pairs of nodes.

$$E = \sum_{i,j=1}^N e_{ij} = -\frac{1}{2} \sum_{i,j=1}^N w_{ij}x_i x_j \quad (2.12)$$

where  $N$  is a number of pairs. The last expression is defined by the fact that the sum includes all the pairs twice. The network usually starts in some initial state and continues the simulation by choosing the nodes in random order. However, there is another possibility that some of the nodes in the network get their outputs fixed and the remainder is to be updated. If the fixed part forms a part of a stable state, the remainder of the nodes will complete the pattern stored in that state. It is similar to the way human brain remembers the things when it is given some partial information on a subject as a hint.

The weights for given stable state vector  $(x^1, x^2, \dots, x^n)$  are determined as follows:

$$w_{ij} = \sum_{s=1}^n x_i^s x_j^s \quad i \neq j \quad (2.13)$$

It should be noted that there is no self activation, which means  $w_{ii}=0$ . The algorithm can be summarized simply as follows:

1. Define the training set and the weight vector

2. Test for desired stable state using the training set to verify the stored stable patterns.
3. Check the energy function for the current iteration
4. Modify the network, energy function and the training set if the result is not satisfactory and repeat the procedure from the beginning.

### Brain-State-in-a-box

Brain-State-in-a-box (BSB) network can be seen as a version of Hopfield network with the continuous rather than discrete and synchronous updating. Apart from this there is no other restriction on the weights. The model consists of a set of neurons or units, which are symmetrically interconnected ( $w_{ij}=w_{ji}$ ) as in a normal Hopfield network and fed back upon themselves. At each time step the units are computed as a weighted sum of the units and this weighted sum is used to update the activation value. A simple non-linearity is added so that the activation value of each neuron remained bounded between min and max values. The state of neural network is represented as a pattern of activation over the neuron units, which is amplified if the activation pattern is 'familiar' to the net and rejected otherwise (Golden, 1993).

If we have  $X$  input patterns with  $D$  dimension, every activation pattern (activation level, consequently firing rate) over model neurons is trapped into a 'box' in  $D$  dimensional hyper region bounded by  $[+1;-1]$  or minimum or maximum activation values. Each of the model

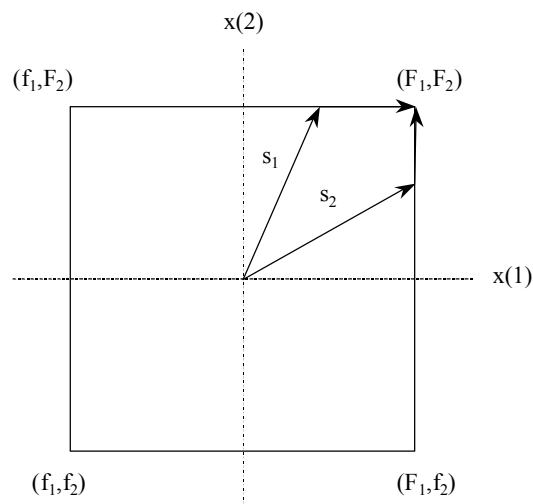


Figure 2.9: Two dimensional state-box (Golden, 1993)

neuron simultaneously adds a weighted sum of inputs and outputs and a bias to its current activation value. In case the range of min and max values is exceeded, it is truncated to the max and min values correspondingly. For example, Figure 2.9 illustrates the two dimensional case and how two distinct system states ( $s_1, s_2$ ) can be mapped into the same hyperbox vertex ( $F_1, F_2$ ).

The learning in the BSB model is formulated as follows:

$$x_i(k+1) = S_i[x_i(k) - \gamma r_i h_i(k)] \quad (2.14)$$

where,

- $\gamma, \rho$  - positive scalar constants
- $S$  - sigmoid activation function

$\mathbf{h}_i(k)$  - net input  
 $x_i(k)$  - activation level or real-valued scalar state

The net input should be defined as follows:

$$\mathbf{h}_i(k) = - \left[ \sum_{j=1}^D w_{ij} x(j) + \sum_{k=1}^D w_{ki} x(k) + b_i \right] \quad (2.15)$$

where,

$b_i$  - bias weight  
 $w_{ij}$  - weight connecting unit  $i$  to unit  $j$  in the model

### 2.3.5 Self-Organising feature maps and other cluster analysis techniques

Self-organising Feature Maps (SOFM) is another type of neural network developed by Kohonen (Kohonen, 1977) which acts upon the theory of associative memory. In associative theory, pairs of patterns are stored so that presentation of one of the patterns in a pair directly evokes the associative pattern. This type of neural network is mostly suitable for pattern recognition and classification.

SOFM uses an unsupervised learning algorithm to map high dimensional data into one or two or at the most three dimensional data space by preserving the key features. SOFM consists of two main parts: input layer and output map (Figure 2.10).

SOFM transforms the input data of arbitrary dimension into a one or two-dimensional discrete map subject to a topological (neighbourhood preserving) constraint. The feature maps are computed using Kohonen unsupervised learning. This network's key advantage is that the clustering produced by the SOFM reduces the input space into representative features using a self-organizing process. Hence the underlying structure of the input space is kept, while the dimensionality of the space is reduced.

Pattern recognition can be defined mainly as a process of automatic derivation of logical pictures of facts underlying the problem domain by categorisation and identification. They work with two steps: feature extraction and the classification. The feature is a kind of measurement taken on the pattern to be classified and transformed into the real numbers. The selected features should provide the characteristics of the input type to the classifier (Beale and Jackson, 1991).

Suppose we have a measurement in the input pattern, each of which is a unique feature. The set of these input vectors is called as a *feature vector*. The dimensionality of the vector  $n$  defines the  $n$  dimensional feature vector. In the data set in the form of  $X$  matrix ( $k \times n$ ) each row contains  $n$  variable values for each of the  $k$  object under analysis. The feature vectors are fed into the Kohonen network, which consists of two layers. The weight values assigned beforehand between the input and output nodes are modified every time the data vectors are loaded to the network. In the end the weight vectors' convergence adopts the information that is provided by the input. The learning procedure mainly takes place within the weight elements and especially the input layer is used just to pass the input.

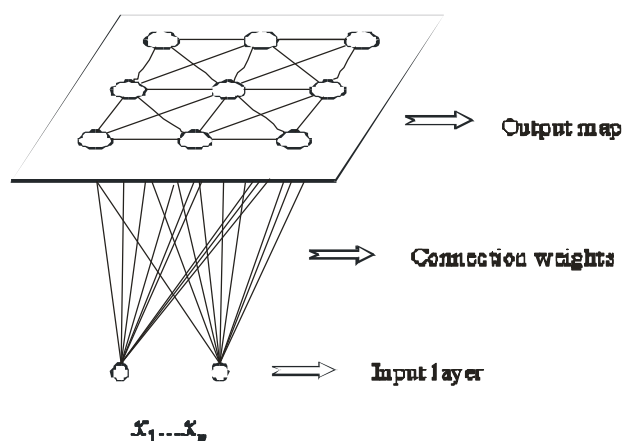


Figure 2.10: Basic structure of SOFM (Beale and Jackson, 1990)

The learning procedure in Kohonen map is a competitive learning. Only the winning node and its neighbouring nodes are updated during the learning and the winning node is considered to be fired. The winning output node is determined by a *similarity measure*, which can be Euclidean distance measure or dot product of two vectors. The best match or the minimum distance measure obtained from the comparison of the input vector and defines the winning node in the output layer. The weight vectors are updated for the winning node and its neighbouring nodes proportional to the value of the difference between the input and weight vectors and a *neighbourhood function*. The learning procedure continues until no significant changes occur in the feature map.

The learning algorithm can be described as follows (Beale and Jackson, 1990):

1. Assigning the weight values. The equal number of weights to the output node is assigned with small random values. The dimension of the weight vectors should be the same as the equal dimension of  $n$  input vector.
2. Input vectors are presented to the network sequentially.
3. The similarity measure has to be done in order to find the winning neuron. As mentioned before it can be either the Euclidean distance measure or the dot product of two vectors. Winning node will be the one which shows the greatest similarity with the input vector. The Euclidean distance that is mostly used for the similarity measure is calculated as follows:

$$D_{lk} = \sqrt{\sum_{n=1}^N (x_{kn} - w_n^l)^2} \quad (2.16)$$

Input vector  $X$  has  $K$  patterns and the number of elements of weight vector is equal to the number of the processing elements in the output layer. In case the Euclidean distance measure is used, the weight to the node which minimises  $D$  most will be the winning node.

4. The values of weight vectors are updated for the winning node and its neighbours. The weight vectors should be updated according to the learning algorithm as follows:

$$\bar{W}^l(t+1) = \bar{W}^l(t) + h(t)N(c,r)[\bar{X}_k - \bar{W}^l(t)] \quad (2.17)$$

where,

- $t$  - number of iteration
- $c$  - number of cycle
- $h(t)$ - learning rate
- $N(c,r)$ - neighbourhood function
- $r$ - neighbourhood radius

5. Repeat all the steps starting from 2 till the network reaches the stable weight configuration. But the term of iteration and cycle have to be distinguished, one iteration is fulfilled when all the input patterns are presented to the network. On the next iteration the input vectors are presented to the network again one by one. But the cycle is the measure of how many iterations has to be done for one neighbourhood radius. It is defined by neighbourhood reduction factor. For example, for the feature map with 8x8 with the reduction factor defined as 2, there will be 5 cycles (8,6,4,2,0).

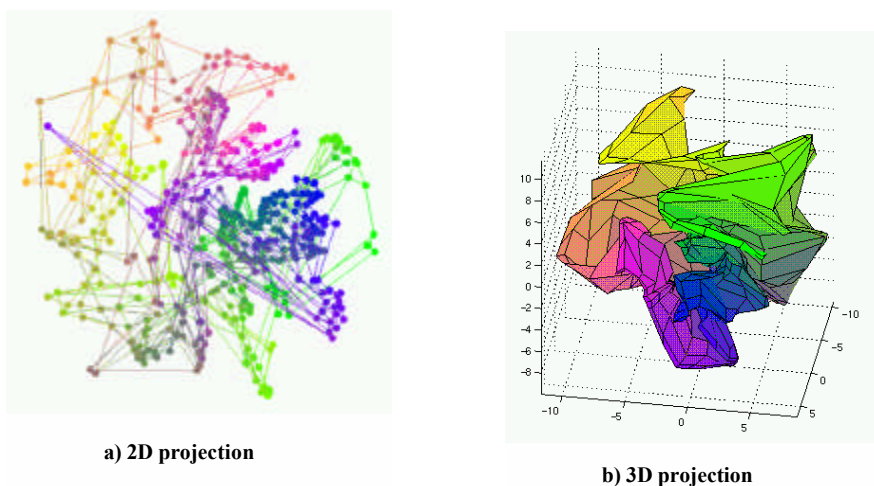


Figure 2.11: Weight vector projection (Vesanto, 1998)

After the training, the result will be the clusters formed by their spatial closeness and this feature is reserved in the weight vector. The analysis of the result of the SOFM is not as simple as in other types of networks. The analysis can be done by visual inspection of the output and the weight connections in 2D or 3D. Different options are available for visual inspections (Vesanto, 1998):

- Count maps, which is the easiest and mostly used method. These maps are formed by winner counts for each output node for the entire data set and it can be interpolated into colour shading as well. The node, which represents the high rate of activation or firing, represents the cluster. It can be plotted in three dimensions in order to make clear visualization.
- Vector position or cluster maps, on which the position of winning nodes for each input vectors are displayed by processing the weight vector. Figure 2.11 shows the projection of the weight vector of the trained SOFM into different dimensions. Colours are coded according to their similarity in the input space. Each dot in Figure 2.11.a corresponds to

one output map unit and the reader can see the concentrations of dots at several locations. Each map unit is connected to its neighbours by line. However, a projection in 3D expresses the phenomenon more clearly than a projection in 2D as shown in Figure 2.11.b.

The distance matrix, which is an Euclidean distance of each output unit to its immediate neighbouring units, can be used for visualization. Then the colours and sizes of representation can be assigned to units with similar property. Minimum, maximum or median of distances between particular unit and its immediate neighbours as well as other cluster properties such as correlation for each pair of units can be used for map visualization. For example Figure 2.12 shows the visualization where the distance matrix is used. In Figure 2.12.a the size of each output unit on the map is proportional to the average distance from its immediate neighbours, so that one can distinguish the different clusters by looking at the areas separated by the larger hexagon. In Figure 2.12.b the same information is being transformed into colours, which means that the similar colours are close to each other in the input space. For more details see Vesanto (1998).

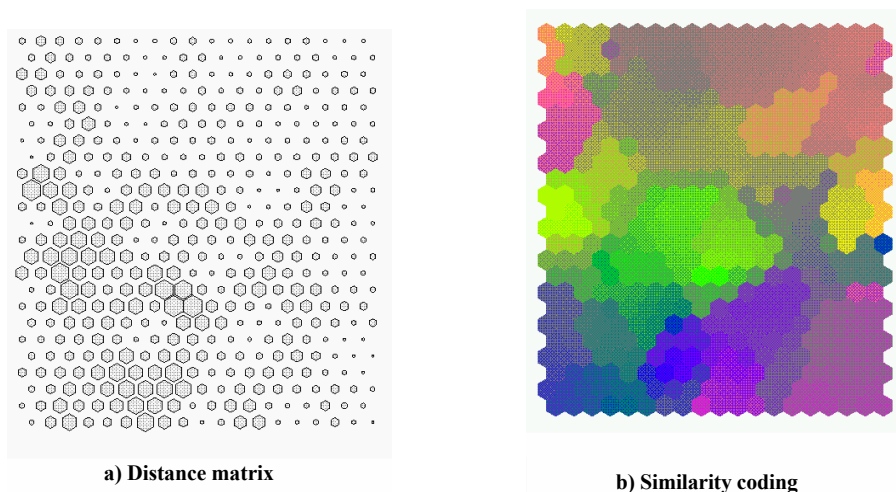


Figure 2.12: Distance matrix for visualization (Vesanto, 1998)

Other than SOFM, there are other numerical and statistical cluster analysis and classification techniques available, such as:

- Adaptive Resonance Theory, which is classified sometimes as a unsupervised, vector-clustering, competitive learning algorithm like SOFM. It also consists of two layers (comparison and recognition layer) and the learning process is executed winner-takes-all manner. The main difference relates to the fact that the classification can be dynamic and there is extensive feedback mechanism between the input and the output layers. When a completely unknown pattern, which cannot be classified into existing clusters, is presented to the network, it can create a new cluster (see Schalkoff, 1997 for details).
- Fuzzy clustering technique, where clusters are defined as a subset of a dataset. Usually the data sample belongs to only one class or cluster but in fuzzy clustering technique sample can belong to several clusters by having different degree of membership to each of them.

- Nearest neighbour method, where any new sampling point is classified to the class to which it is located closer in the Euclidean space. In the input space, the classes are determined by certain radius defined beforehand. The method is applied better if the patterns are well separated (see Beale and Jackson, 1990 for details).
- 
- Optimization partitioning, which classify the data set into groups by optimizing some predefined criterion. The difficulty occur to find the global optima of the defined criterion. As an example, support vector mashines algorithm can be mentioned. (Vapnick, 1995)
- 
- C-means clustering, which improves the homogeniety within the groups by minimizing the sum of squares of the patterns. After carrying out an iterative procedure, a certain number of cluster centres are found.
- 
- Linear classifier method, which is applied for linearly separable problems. The linear decision boundary to separate classes is defined by descriminant function. Value of discriminant function is used for classifying given sample pattern on the basis of the weight vector (see Beale and Jackson, 1990 for detail).

### 2.3.6 Principal component NN

This type of neural network is based on the standard or linear Principal Component Analysis (PCA), which is a multivariate data analysis technique used for data reduction in terms of dimensionality and noise reduction (generalization). The standard PCA technique is in fact an orthogonal coordinate transformation (Diamantaras and Kung, 1996).

In PCA,  $n$  dimensional redundant data vectors, which are correlated to each other, are transformed into certain  $s$  dimensional data vectors ( $s < n$ ) *orthogonal* to each other. These vectors give the principal directions along which the data cloud mostly stretched. The

principal components are the projection of the data set on eigenvectors of principal directions (Figure 2.13). This is equivalent to drawing a regression line through the points described by the original data pairs, and describing a new point as the distance along that line (Swingler, 1996). The eigenvalues for given eigenvector give an indication about the information of that particular principal component. In other words, the first principal components ranked with their eigenvalues in descending order explains the most variance of the data set and the last explains the least.

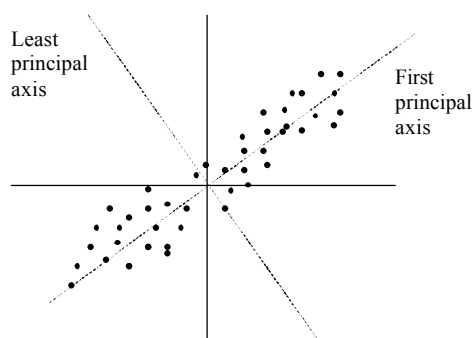


Figure 2.13: Linear principal component showing the most and least variance of data set (Diamantaras and Kung, 1996)

By ignoring the last principal component, data reductions can be made as long as the data set still contains the most valuable information. Principal Component Neural Networks (PCNN) are mainly used for classification and feature extraction. The reason is PCA makes it easier for classification by extracting the most important information for classification and removing the correlation between the attributes, which possibly slows down the identification of the classes. PCA and PCNN are explained in a bit more detail below.

### *Principal component analysis*

The PCA analysis of a data set in fact is a rotation of given data vectors. The new axes as a result of the PCA analysis should form the principal directions in the orthogonal data-space. Let us assume that the data set of  $k$  observations of  $n$  variate variables that can be represented in a form of  $X$  matrix ( $k \times n$ ). The algorithm of transforming data into uncorrelated set using standard PCA can be formulated as follows:

1. Calculate the covariance matrix of the data matrix. In order to find it, the original data matrix should be centred around their mean value first. It can be obtained by calculating first the mean and their variance:

$$\bar{x}_n = \frac{1}{K} \sum_{k=1}^K x_{kn} \quad (2.18)$$

$$s_n = \sqrt{\frac{1}{K-1} \sum_{k=1}^K (x_{kn} - \bar{x}_n)^2} \quad (2.19)$$

Then the covariance matrix is calculated from the centred matrix  $Y$  as follows:

$$S = \left( \text{cov ar}[y_i, y_j]_{i,j=1}^N \right) \approx \left( \frac{1}{K-1} \sum_{k=1}^K y_{ki} y_{kj} \right)_{i,j=1}^N = \frac{1}{K-1} Y^T \cdot Y \quad (2.20)$$

2. Calculate the eigenvectors of the obtained covariance matrix, which should fulfil the following condition:

$$S \cdot X' = I \cdot X' \quad (2.21)$$

where  $X'$ - eigenvector and  $\lambda$ - eigenvalue of the covariance matrix

The determined eigenvalues should be the fraction of the total system variance. The eigenvectors orthogonal to each other are principal directions.

3. Projection of data set on the eigenvectors determines the principal component. Which means the transformed data set can be obtained by multiplying the original matrix by the eigenvector matrix.

$$Z = X \cdot A \quad (2.22)$$

where  $X$ - is original data matrix and  $A$ -is columnwise eigenvector matrix



### Principal component neural networks

Sanger and Oja proved that one-layered linear neural network is equivalent to the linear standard PCA. And the neural networks which implement this learning algorithm is called Principal Component Neural networks (PCNN). For details see Diamantaras and Kung (1996). If the network has  $p$  inputs (or given sample has  $p$  components) and  $m < p$  outputs, the learning named as Hebbian is as follows:

$$y_j(t) = \sum_{i=0}^{p-1} w_{ij}(t)x_i(t) \quad (2.23)$$

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha y_j(t) [x_i(t) - y_j(t)x_i(t)] = \\ &= \alpha \left[ y_j(t)x_i(t) - y_j(t) \sum_{k=0}^j w_{ki}(t)y_k(t) \right] \end{aligned} \quad (2.24)$$

where  $\alpha$  is the step size

The Hebbian rule is characterized with involving the product of a pair of node activation or outputs. By doing this procedure we are calculating the eigenvectors of the correlation function of the input without computing the correlation function. It was also shown that this learning converges to the solution where the weights of the PCA network approach the first principal component of the data matrix. Also by this learning procedure one can reduce the dimensionality of the original data set ( $m < p$ ).

Disadvantage of using the PCNN remains in the outlying data points, which distort the eigenvalue estimation and causes skewed data projections. Also if the classes are on the top of each other it does not guarantee of the class separability. If the PCNN is used for dimensionality reduction purposes, it may destroy the non-linear relations attempted to be modelled.

## 2.4 Applications of Neural networks in the water sector

### 2.4.1 Drinking water systems

#### Prediction of drinking water consumption

Prediction of community water consumption is not an easy task, there might be many influencing factors on the subject. Only few water supply companies in the Netherlands use automated prediction models. Moreover, the accuracy of the prediction by these models is not always satisfactory, particularly during the peak consumption period on the daily basis, the error level reaches 25%. Aafjes *et al* (1997) investigated a short term prediction of community water consumption by ANN, traditional expert system and combination of ANN and expert system. Water consumption data of Friesland for two years have been used for the study.

For neural network model development, 5 variants have been studied to relate the predicted water consumption to the previous 1 to 7 days' consumption data. The hourly consumption data for previous day and hourly consumption of the same day one week before, together as input variables gave the best result. Also the day of week is given as input because the water consumption may vary on different days of week. Obviously the climate characteristics are one of the influencing factors for water consumption. Therefore, the measured data such as air pressure, global radiation, temperature and the precipitation are included as well. The climatic data during daytime, between 9.00-19.00 hours improved the network performance. The inclusion of difference between global radiance of the current day and previous day significantly improved the performance. Neural network models allowed decreasing the prediction error by two times.

Expert systems have been developed for the days with big prediction error of neural network: national holidays, school holidays and day after holidays. Before developing the rule base for expert system, the fault analysis has been carried out. The case based reasoning of expert system is used for fitting the predicted water consumption by neural networks. After making neural network prediction, the expert system is used for selecting the same holiday in the past from the case library. The difference in water consumption is used for correcting the error of neural network output. The error is corrected up to 75% by use of case based reasoning.

The accuracy of neural network based prediction of water consumption is considered as fair to good. For short-term prediction, the comparison of ANN model's result with the conventional statistical analysis based model's (ARIMA) result shows an improvement in the ANN model. Climatological data can significantly improve the model performance. If training data includes longer time series then the result can be improved by including more data on water consumption during holidays in the case library of expert system. In this case study, the neural networks are trained off-line. On-line training option must be implemented, so that the neural networks can be trained on the basis of newly measured data.

### Drinking water quality

In water quality control, the estimation of water quality evolution from the treatment plant to the consumer's tap is an important issue. During the water transportation through the distribution network, the residual chlorine concentration guarantees microbiologically safe water quality. The residual chlorine concentration diminishes due to the reactions within the pipeline. The comparative study of conventional first order modelling approach and ANN model on the residual chlorine evolution is carried out by Rodriguez *et al* (1997). The conventional model with first order semi-empirical equation is as follows:

$$C_D = C_U e^{-KT} \quad (2.25)$$

Where  $C_D$  and  $C_U$  denote the chlorine concentration at the downstream and upstream point in the system respectively,  $T$  is the travel time and  $K$  denotes the coefficient of chlorine decay, successful determination of which makes the model reliable.  $K$  may vary in time and space depending on many parameters. Therefore, a lot of work on accurate estimation of the coefficient is needed. The error backpropagation neural network model takes time lagged and delayed values of  $C_U$ , water temperature, water flow rates measured on-line, travel time as input and produces  $C_D$  - downstream chlorine concentration as an output. The travel time is calculated from the first order flow equation by assuming the flow rate as constant with space. The models were built for both steady and unsteady flow conditions taking seasonal

variations into account. The obtained results of ANN model show high accuracy and make the combinations of the two approaches as promising in this particular field of research.

Zhang and Stanley (1997) investigated the problem of forecasting of raw water quality coming to the treatment plant using neural network model. In order to meet the changes in incoming water quality and supply high quality water to the consumers by adjusting the treatment processes in an optimal manner, it is desirable to know the quality of incoming water in advance. By previous research it was found that the colour of raw water and turbidity are the most important parameters to affect the treatment processes. To predict the colour of water, the inputs to the neural network model are turbidity, river flow rate, precipitation at a meteorological station located upstream in the basin and their derivatives. The result of ANN model is found to be promising and it may serve as a solid ground for real-time operation such as computerised coagulation dosing control.

#### 2.4.2 Sewerage systems

##### Floc classification

Classification of floc is a very important issue in water purification process. The floc size and its structure are directly related to the technical process parameters such as dewatering behaviour or settling ability of sludgy wastes (Nagel, 1999). The classification of floc usually is done by numerical clustering techniques, but for analysing their result particularly for overlapping or dense cloud of sample, the efficiency must be improved. Use of neural network classification may be one of the possible alternatives.

##### Water treatment process control

Water treatment process control, especially determining all the different micro scale physical and chemical reactions numerically, which are highly non-linear processes, is a complex task. The mathematical descriptions of the processes give rather limited success in controlling the processes in real-time, therefore the chemical dosage for different stages of treatment is decided by expert's judgement. ANN is used for controlling coagulation-flocculation-sedimentation processes and in determining the optimal chemical dosage on the basis of the water quality parameters of incoming water (Zhang and Stanley, 1999).

In treatment plant, the water quality parameters to represent the state variables are pH, turbidity, colour and temperature. The control variables are chemical doses for alum, Powered Activated Carbon (PAC) and the clarifier overflow rate. Proposed feedforward control scheme with optional feedback loop is shown in Figure 2.14. The control scheme consists of three main parts.

- a) Neural network process reference model, which takes the measured water quality parameters, intended dosage for alum and PAC and produces the turbidity of water coming into the clarifier
- b) Noise filter for calculating, selecting and passing the preferred control information from sources such as process standards or feedback, and the prediction from the previous plant reference model to the next unit. In other words, its purpose is to check whether the predicted turbidity of effluent water matches the standard. Also it enables and disables the feedback flow of information. If the actual turbidity is greater than the reference, then the

difference is subtracted from the desired turbidity. Otherwise the difference is added to the desired value and that value is passed to the inverse model.

- c) Neural network inverse model works completely the other way. It accepts the raw water quality parameters, the desired PAC dosage, predicted turbidity to the clarifier and generates the proper alum doses.

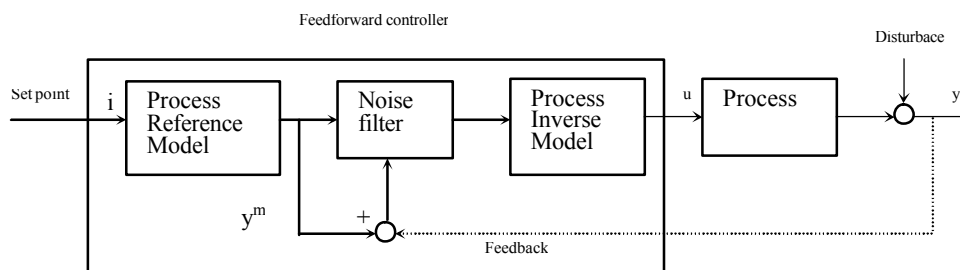


Figure 2.14: Control scheme

$i$  - Intended dosage of alum and PAC

$u$  - the alum dosage

$y$  - output of the process, turbidity

$y^m$  - the turbidity difference

The control algorithm determines the alum dosage to bring the difference between the desired turbidity and actual turbidity to minimum. If the alum and PAC dosages at the previous sampling time step are good for current condition, those are passed to the process reference model with the present water quality parameters to predict the turbidity. The filter unit determines whether the turbidity is above the limit (the alum dose value should be changed) or not (the turbidity is passed to the inverse model to find the corresponding alum dosage). The performance of the control scheme is highly dependent upon the accuracy of two neural network models. It is concluded that the models perform well, however, with less noisy data and implementation of on-line measurement, the performance should be improved.

### Emulation of sewer flows

Simulation of sewer flows in the Netherlands using ANN has been investigated by Proano (1998). Nearly 90% of sewer systems in the Netherlands are combined systems, where the storm water and the wastewater are transferred to the sewage treatment plant with the same pipeline. When combined sewer discharge exceeds the capacity of the treatment plant the excess sewer is discharged directly to the delivered water, causing serious environmental pollution every year. The situation is known as Combined Sewer Overflow (CSO) problem. The data set for training and testing the network is generated by SOBEK-Urban package developed at Delft Hydraulic Institute.

The problem of simulating the sewer overflows was investigated first on simplified scheme and encouraged by promising result, the real sewer system of Maartensdijk town in the centre of the Netherlands was considered. This is a flat and large system with 445 branches, 406 nodes, with three overflow weirs and one pumping station. Rainfall events were chosen to run SOBEK-Urban model under certain assumptions in terms of their duration and depth from measurement data of 10 years. Result of SOBEK model simulation forms a training data set for neural networks. Thus ARNN models were built for each overflow structures taking precipitation and water depth at previous time step to produce the discharge and water level

at the following time step. The resulting water level becomes an input along with the predicted precipitation to produce the water depth for the next time step and discharge at the current time step. The approach has been concluded as 7 times faster than the SOBEK simulation. The result of considering two simultaneous overflows at two different points has also shown a reasonable accuracy. However, further investigation is needed to improve the accuracy of solution.

### Water treatment plant parameter prediction

Another application of error backpropagation network for predicting water treatment plant parameters was investigated by Hanisch and Pires (1998). Determining those parameters at each processing level such as plant input, input to the primary settler, input to the secondary settler and as plant output, allow a better process control and management. The network models take past and present values of 9 parameters such as total water capacity, pH, conductivity, Zn, BOD, COD, settleable solid concentrations etc. as input variables and predict their future values in total of 29 output variables. The error of the obtained result shows somewhat high value for the purpose of the study, and the study should be continued to improve the results.

### Selection of wastewater purification plant type

The problem of determining the class of appropriate wastewater purification plant when a set of parameters of pollutant types and their characterization are given was investigated by Vermeersch *et al* (1999). Modelling of ill-defined problem such as water purification process usually consists of three phases: relationship detection, model structure characterisation and parameter estimation. Structure characterization that is considered in this study is a specification of parts of functional relationships between variables. The property of the approach considered in this study is to identify certain characteristics or features of the system. Features are compared to the corresponding feature of each known class of candidate models. Those candidate models are assumed to be based on the set of candidate structures; the parameters of each structure required to be estimated are given a value within certain limit.

If the feature of the system to be modelled is resembled to one of the structure, then that particular class of candidate models can characterize the system. Therefore, features of a candidate class are input to the network and the corresponding class is an output from the network. The neural network learnt the features of the specific classes and finally, the net returns the suitable technology of water purification process by taking the real-world values. Different techniques and different structures of the neural networks are tested in this study; error backpropagation networks, SOFM, recurrent neural networks, Brain-State-in-a-Box (BSB) models, learning vector quantization and nearest neighbourhood methods. The data for classification are pre-processed beforehand in order to keep the magnitude of the network as small as possible and also to get the data well separated in the input space which prevents overlapping of classes.

Each parameter for wastewater purification process has value a within a certain range that is chosen experimentally. The efficient control of wastewater purification requires a clear view of the components of the influent water - how many pollutants are incorporated in the water, and whether the pollutant is degraded under saturating or non-saturating conditions etc. Training data is obtained by processing measured data by bio-sensors using four different

transformation techniques: Fourier series, fast wavelet transform, Karhunen-Loeve transform and simple decimation technique.

In total 23 data patterns, each of which consist of 600 data points, are used as training data for classification. The result of neural classification is compared with the classification of real-life data classified by human experts. Within the 5 different classification methods the BSB model and backpropagation networks achieved the best result using the simple decimation method and fast wavelet transformation methods for data preprocessing. Therefore, it has been concluded the data preprocessing is essential for better functioning of neural network. The BSB models have advantage of having their architecture fixed and not having a training phase so that there is no danger of overtraining. Also all the neural networks proved to outperform statistical classifiers such as nearest neighbourhood method.

### Sewer water quality

Models for water quality simulations are normally based on a number of simplifying assumptions about the process and they need a lot of effort for calibration in order to get an accurate result. Nouh (1996) used error backpropagation network to simulate the sewer water quality using measured data for rainfall duration and intensity, the catchment characteristic data, and pollutograph, which is suspended solids, nitrates, total phosphates, total particulate suspended solids concentrations. Pollutographs used for NN simulation and verification were generated by SWMM model after proper calibration. The NN accuracy is satisfactory only for simulation of peak suspended solid concentration in small catchments. Also the proportion between the depth and duration of storm events and spatial and time variation of the storm event have to be considered carefully.

### Waste water treatment process

Zhao and McAvoy (1996) reported the result of application of neural network and First Principle Method (FPM) combination for activated sludge processing problem. FPM is same as the Principle Component Analysis (Section 2.3.6). Activated sludge technology is used widely in waste water treatment plants, which is a very complex bio-chemical process. The models describing the dynamics of the process are available, however, the calibration of number of variables and parameters is costly and time consuming. The authors assumed that the lack of on-line measured data makes it preferable to use a control structure of neural network combined with the FPM.

The control scheme (Figure 2.15) implemented should take plant inputs such as raw water BOD, total nitrogen, ammonia nitrogen, and external data such as temperature, pH etc. and produces the parameters (BOD, and other nitrogen compounds) at the effluent to the clarifier. The FPM part produces the parameter values at the effluent from the plant inputs only, then the neural network part takes all the plant input, external disturbance data, the time delayed inputs and the residual value. The residuals ( $e(t)$ ) are the differences between the desired value of plant output and FPM output, which is an error of FPM. Once the training is finished the switch is turned to the position P and outputs of the network are fed back with time delays ( $z^{-1}$ ) to the neural network. Then output of the NN is the residuals. Finally, the control system output is the summation of FPM and the residuals predicted by the recurrent neural network model.

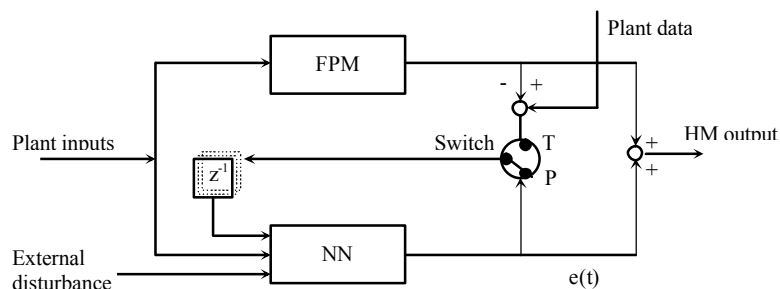


Figure 2.15: Hybrid model structure

For NN part of the scheme, an error backpropagation network and a recurrent neural network are used. The scheme was tested with both simulated and measured plant data. Combining the neural network to FPM, which is accurate for steady state, makes it possible to make predictions in process dynamics and add the external data. The resulting accuracy of the two combined techniques is explicitly higher than the singular models. Moreover, the proposed approach significantly reduces the task of calibrating the FPM and the hybrid modelling approach can be used for other complex processes as an accurate and cost-effective modelling tool.

### Urban runoff

Loke *et al* (1997) studied application of neural networks for prediction of runoff coefficient by using simple catchment data, while regression model for this sort of task requires rarely available data such as soil moisture deficit or soil structure. The input data for error backpropagation network consists of conventional catchment characteristics such as catchment size, percentage of impervious area, which can be easily derived from normal topographic maps. The percentage of pervious and impervious area and the sum of impervious and semi-pervious area are found to be the most suitable input variables. The result of verification illustrates the prediction error within 10-20% range, with average deviation of about 4%. If the number of training example is sufficient the performance should be improved.

The authors applied neural networks for filling in gaps of measurement data as well. By the measurement of two rain gauges in Copenhagen, the measurement of the third one was restored. The result is compared with the simple substitution method and a satisfactory result obtained illustrates the neural network's ability to deal with this type of problem.

### Urban runoff prediction

SinCak *et al* (1998) used Radial Basis Function (RBF) network and Cascade Correlation (CC) networks for predicting the sewer flow on the basis of historical rainfall data. Data for sewer flows are continuously measured by ultrasonic level sensors at three cross-section points in the sewer system. Rainfall data is measured from the gauge in the town centre. Different network topology was investigated with time lagged or moving average values of rainfall data as input and sewer flow as output. CC neural networks, which is a special type of error backpropagation networks showed higher performance above RBF network for prediction of sewer flow ahead. The advantage of CC neural networks is that it optimizes the topology by itself.

### Urban storm drainage

Storm water is usually collected by the sewer system and is discharged to the rivers after the treatment, in order to prevent the washing up of pollutants such as heavy metal, hydrocarbon, micro-pollutants etc, from the impervious area. The optimal management of the treatment plant consists of minimising the operational cost while assuring the quality of discharged water. A STORMNET connectionist model was built with two specific recurrent neural network models to simulate runoff and solid transfer in the sewer system (Gong *et al*, 1996). Data was generated by HYPOCRAS conceptual model.

The first part of a model uses the rainfall intensity and accumulated rainfall depth and generates the effective rainfall intensity. It is necessary to determine the effective rainfall intensity because the runoff forms only after attaining certain level and not all the rainfall turns into runoff. Then the effective rainfall is used to produce the flow rate by recurrent neural network model.

The second part takes the flow rate and produces the solid transport with some simplifying assumptions. STORMNET was tested for different urban catchment sites of different size and for different intensity of rainfall events, the results are very promising and accurate. It is concluded that for a larger catchment the number of rain gauges to collect data has to be increased.

### Control strategy selection system for urban drainage

The main goal of real-time control of urban drainage system is the full utilization of the existing infrastructure and resources by satisfying the operational objectives. Operational objectives are multiple in nature such as reducing operation management cost, equalizing treatment plant inflows, reduction of surface flood or Combined Sewer Overflows (CSO), consequently reduction of environmental effects due to flooding etc. Possibly the objectives are conflicting with each other. Wilson (1995) investigated the suitability of a rule-based learning classifier system technique for urban drainage system control.

The considered technique in this study as a basic learning mechanism is a so-called Q-type learning. Each rule maintains Q function, which calculates an estimated future cost for given state and action. The function can be formulated as follows (Kavehercy, 1996):

$$Q^n(x, a) = q(x, a) + \gamma J^n(y), \quad (2.26)$$

where

$y$  - state resulting from applying action  $a = u(x)$  to state  $x$ ,  $J^n$  - evaluation function and  $\gamma$  - discount factor

The action decisions are based on cost prediction, where the minimum cost will lead to the objective achievements. The system learns from hydraulic simulation. During the learning phase, rule that *matches* the event provides the future cost predictions, and if an event has no set of match, then the classifiers are generated for each possible action. The classifier system is considered to be available for application to many types of engineering control problems, and it does not suffer from simplified constraints. Genetic algorithm is used in this technique for background rule induction and updating. For data generation MOUSE system of DHI is used. The technique has been successfully applied to a large urban drainage system.



However, the learning itself demands large computing power in order to get a desired accuracy.

### 2.4.3 Inland water systems

#### Predicting water level

Problem of predicting water level and the delivery amount from the low lying polder areas to the alien water has been investigated on the specific case of South Holland province water authority (Lint and Vonk, 1999). The water authority is responsible for three distinct regions, from which excess water is delivered to the Lek River. One of the regions, named as Overwaard region, comprises of 22 polders and has to discharge water into one low lying reservoir, which in turn discharges to another high lying reservoir. Finally, the high lying reservoir discharges to the river by sluice (Figure 2.16). There is a possibility of minimizing the cost of energy consumption by pumping during the night hours (between 23.00-7.00). In order to do that it is important to know the following parameters 24 hours ahead:

- delivery amount from each 22 polder areas
- expected water level in the high lying reservoir
- expected water level of the Lek river at the sluice gate

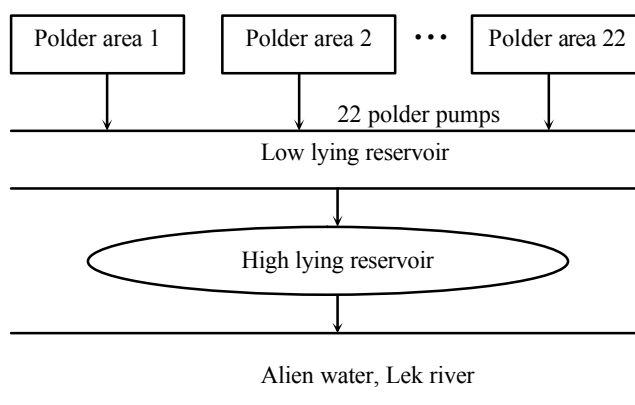


Figure 2.16: Scheme for water delivery (Lint and Vonk, 1999)

The expert system developed for above purpose is considered to be inaccurate, therefore neural network technique (MLP) is investigated as an alternative methodology. The training data set is built on the basis of the SCARK database, which is an automatically operated measurement system. Two distinct neural networks were built with the following inputs with 1 hour time step:

- water level for preceding 12 hours
- precipitation for preceding 12 hours
- temperature for preceding 12 hours
- pump status for preceding 12 hours
- predicted temperature for 1 hour ahead
- predicted precipitation for 1 hour ahead

Water level and pump status for the following hour is obtained as output of the neural network models. For normal situation the error backpropagation networks and for extreme precipitation situation the radial basis function networks were found to be suitable. The obtained result is considered to be satisfactory and the model is being applied for

Rijkswaterstaat. However, more input variables such as upstream water level and precipitation data at the Lek river have to be included.

### Prediction of lake water level

Auto Regressive Neural Network (ARNN) has been applied for predicting water level of the lake IJsselmeer at the North-Holland on the basis of incoming river discharge, water level at the sea-side of the sluices and wind event (Gautam, 1999). Although precipitation over the lake and evaporation influences the lake water level, the effect is negligible compared to other parameters mentioned above. The lake discharges water through the sluices during the low tide and it is important to know the water level and the amount of discharge to the Sea. The storage and discharge of the lake, wind speed and direction, water level at the sea side, the daily low tide water level are considered as inputs to the networks. Trained on the measurement data, ARNN has been found to be a promising tool to predict the water level, showing a slightly better result than the numerical modelling technique.

### Classification of river discharge patterns

SOFM is used for classifying the discharge patterns of Mekong river using time series data (van Boogaard *et al*, 1998). Classification is based on the normalised 62 years of data (patterns) with 12 dimension (monthly discharge). Normalisation has been made with special attention so that:

- patterns getting the residuals with respect to the yearly average discharge
- components having the same spread.

As a result, 4 distinct classes of discharge patterns were found: dry years, wet years, dry years for the first half and wet for last half and finally wet years for first half and dry for last half.

### Identification of pollutant source

Götz *et al* (1998) carried out comparative study on identification of possible sources of overwhelming dioxin contamination in the river and harbour sediments using Kohonen neural network and multivariate statistical technique. The data used for classifications are samples from sources of possible contamination along the river reach such as sludge processing, pesticide factory copper slag and were collected in the form of soil samples, air samples, sediment and suspended particle sample, surface water sediment, samples from the flood plain etc. The preprocessed sample data consists of 407 exemplars, each of which contains 18 different parameters.

By analysing the clusters formed, it is possible to identify the sources of contamination for each subreach. Clusters formed by 2 approaches were nearly identical, however, Kohonen networks give more consistent classification and proved its high potential for classifying the environmental data and identifying the source of contamination.

### Water quality management in river basin

Water quality management in river basin is a multiple objective decision making focusing on goals to find a reasonable allocation of waste loading for each pollutant source, to enable environmental quality for living organisms in the river by determining the maximum possible mass loading to the river. However, the problem does not give an appropriate solution as long as the solution is based on the decision maker's preference, which is necessary but is always

ill-defined in all planning procedures. In multiobjective optimization, high non-linearities exist between the values of objectives and their relative weights, because decision maker's preferences may not be clearly defined. At the same time, multiple and non-commensurate objectives are difficult to classify in terms of their priorities and weights. Also sometimes different objectives could lead to the same weight combination. The ANN application for prediction problem of decision maker's preferences in the objective-weight relationships was studied by Wen and Lee (1998).

The study focused mainly on the environmental quality, treatment cost of wastewater, assimilative capacity of a river to provide a solution to water quality problem in the basin. Also the study is based on the method of minimising the distances of the real and the ideal objective solutions within the feasible region by using payoff table. Furthermore, the following compromise programming model is solved by non-inferior method:

$$\begin{aligned} & \text{Min } d \\ \text{Subject to: } & \mathbf{p}_k (Z_k^M - Z_k(x)) \leq d \quad k = 1, 2, \dots, p \\ & d \geq 0 \\ & x \in X \end{aligned} \quad (2.27)$$

In which  $d$  the maximum weighted deviation of each objective from the ideal solution,  $x$  is a vector of decision variables,  $X$  is a feasible region,  $Z_k(x)$  is the  $k$ th objective function,  $Z_k^m$  is the maximum of the  $k$ th objective found from the payoff table and  $\pi_k$  is the  $k$ th weight indicating the relative importance of the deviation that may be indicated by the decision maker's preference.

The optimization scheme proposed in this study consists of a neural network, which produces the weights of the objective function. This is trained on the decision maker's preference database and multiobjective optimization phase. Data sets to train the error backpropagation network were developed by random generation from the database. In this case study, there are 3 objectives and constraint sets: minimize BOD concentration of the first reach of the river, minimize the water treatment cost for whole river basin and maximize the total allowable loading rate to every reach of the river. The constraints of optimization are the water quality standards, water quality model results and the equitable removal of wastewater.

The neural network takes the actual values for all three objective functions and it produces the suitable weights for each objective function. Then the weights are used for obtaining the non-inferior solution from the general compromise optimization problem. The result concludes that the neural network model based multiobjective optimization approach can be a powerful and promising tool for water quality management in a river basin.

### Controlling polder water level

The problem of using AI techniques such as neural networks and Fuzzy Adaptive Systems (FAS) for Real Time Control (RTC) of regional water resources system was addressed by Bhattacharya (1998) and further by Bazartseren (1999). In this study Aquarius DSS model (Lobbrecht, 1997) was used as a reference model in model reference adaptive control. Neural network trained in off-line mode and FAS tools were used for reproducing the simulation results of Aquarius model, in determining control actions for regulating structures (see figure 2.11).

The MRAC is supposed to work as a conventional feedback control scheme. The desired value  $y(t)_d$  or target water level in the polder area passes through the intelligent controller and gets the control signal  $u(t)$  or pumping rate of the drainage station, which results the system output  $y(t)$  or water level in polder area (Figure 2.17). The resulting water level then should be compared back with the target value. The output error should be manipulated through the controller and a new water level is obtained in the polder area through the redefined control actions.

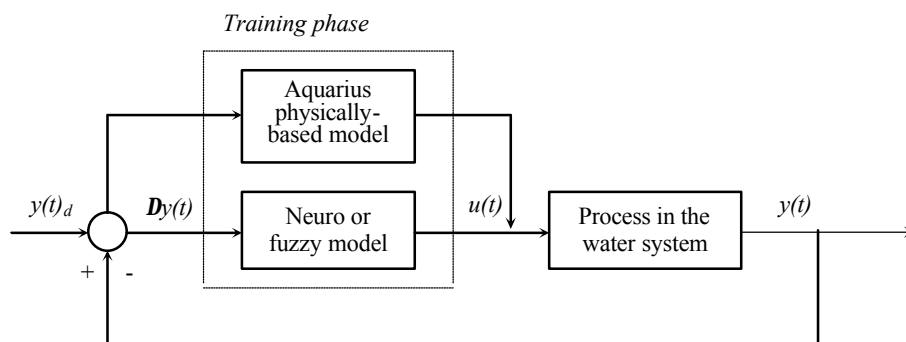


Figure 2.17: Model Reference Adaptive Control scheme (Bazartseren, 1999)

The input variables for intelligent controllers were chosen by pre-processing of data. The accuracy of control actions replicated by neural network and FAS are satisfactory and comparable for not only local control but also for centralised dynamic control mode. Therefore it was concluded that ANN and FAS could make the use of Aquarius DSS practical for RTC. The study is concentrated on one type of regulating structure, which is pumping station. Problem of controlling water levels by determining the suitable pumping rate in several different water resources system models in the Netherlands were considered. Further investigation of this approach is suggested in terms of improving the performance by selecting suitable state variables as inputs.

The study further developed to use the trained ANN or FAS parallel to the Aquarius DSS (Lobbrecht *et al*, 2000). It implies that the intelligent controllers are implemented into the control loop to produce the required control actions on-line. The result proves their ability to reproduce the local and centralized control actions on the basis of locally measurable information only. However, the study should be extended to investigate the application of intelligent controllers into larger water system models and also for other types of control structures.

#### Control strategy in multi-reservoir system

Determining the quasi-optimal control strategy for a multi-reservoir system, using error backpropagation network was addressed by Solomatine & Torres (1996). The optimal use of water resources within river basin (Apure river basin, Venezuela) had to be achieved by increasing the navigable period without decreasing the energy production. The hydrodynamics and hydrology of the basin was simulated using MIKE-11 modelling system. In order to solve optimization problem in a shorter time the model result was approximated by neural network generator NNN. And NNN generated code for independent run was used in optimization loop. The multi-objective problem was formulated as dynamic programming problem taking a navigability constraint as a 'soft' constraint and it was first solved with only one control variable - energy release of a reservoir. The problem is solved by increasing the non-energy release (bottom outlet release) from zero at every run of optimization problem

until the water level along the river reach becomes high enough for navigation. Proposed scheme for deriving optimal control strategy is shown in Figure 2.18. The proposed approach of model approximation was suggested for various schemes of water resources optimization.

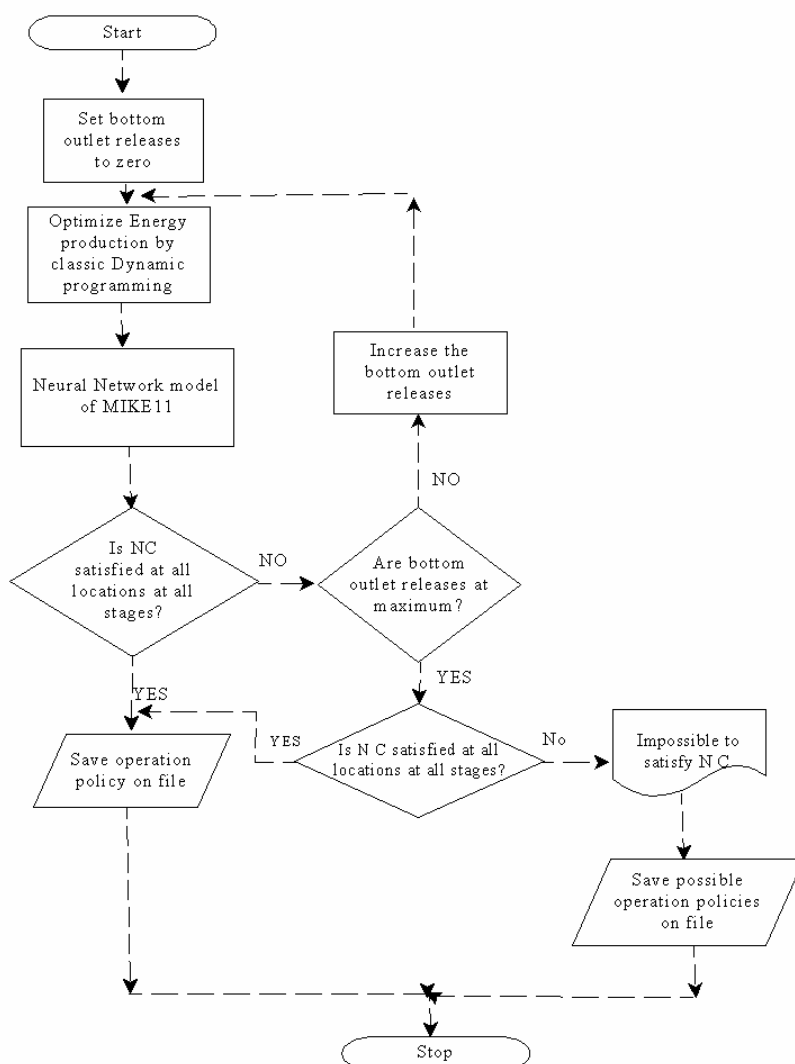


Figure 2.18: Scheme for deriving optimal control strategy (Solomatine and Torres, 1996)

Raman & Chandramouli (1996) used Multi Layer Perceptron (MLP) for a similar problem of determining control strategy in reservoir system. The operation policy was determined by solving dynamic programming, stochastic dynamic programming algorithms and linear regression procedure. The dynamic programming problem of minimizing the squared deficit of the release from the irrigation demand was solved for 20 years of historical data. Also 10 years of meteorological data was used to find the irrigation demands. The neural network model was trained on the basis of data generated by dynamic programming. The inputs to the neural network model were initial storage, inflows and demands. The results from NN were compared with the result from stochastic dynamic programming. The standard operation policy determined by linear regression. The three-year's simulation performance of all four models were compared and the squared deficit obtained by ANN was the least within these models. It was concluded that the use of neural networks for this kind of problem has a potential and has to be investigated further.

Application of ANN for optimal reservoir operation has been investigated by Lee (1997). ANN is applied as a part of the optimization model and it has been used for approximating flood routing model. The obtained result has made the real-time reservoir operation easier.

### Non-linear ecological relations

Cyanobacteria (algae bloom specie) mass is one of the criteria of deterioration of river water quality and significant percentage of these blooms are found to be toxic (Maier *et al*, 1997). Formation of these species is not well understood, therefore, it was an appropriate area of using ANN as a predictive tool for incidence of cyanobacteria. The research focused on estimation of effects of water quality variables to the growth of cyanobacteria and the possibility to forecast the species' growth 4 weeks in advance. The data used for prediction of the bacterial growth is species' population, watercolour, turbidity, temperature, daily flow, phosphorus, total iron and oxidised nitrogen in the water.

ANN models were able to predict the population of cyanobacteria with above water quality variables with high accuracy (average of 325 cells/ml). ANN model also could predict the most important variable that has greatest effect on timing and the incidence of the specie. The most influencing variables were found to be colour representing light attenuation and light availability rather than other chemical substance concentrations.

The similar application of ANN for prediction of algae blooms on the basis of water quality parameters for 4 different freshwater systems was studied by Recknagel *et al* (1997). The output of the network models was 10 different species of algae blooms and different water quality variables were obtained to be the most influencing variables for each freshwater systems. The study revealed ANN's ability to model very complex non-linear ecological phenomena.

Lek *et al*, (1996) used error backpropagation networks to identify the non-linear relations between the physical habitat variables and the density of a brown trout redds in the stream bed. The habitat variables are measurable variables such as wetted stream width, surface velocity, water gradient, mean depth, bottom velocity, area with suitable spawning gravel for trout etc. The output is a single variable representing a density of brown trout spawning redds per unit length of the stream on the basis of 6 mountain stream data. Performance of a neural network model is compared with the stepwise multiple regression analysis result. For regression models, the variables have to be transformed, however, neural network model performed better than the regression model with raw data.

The comparison of ANN model and Regression Model (RM) in predicting attributes of terrestrial ecosystems by temporal prediction of functional attributes at regional scale was addressed by Paruelo and Tomasel, 1997. Total of 6 functional attributes are predicted using generated data from the seasonal course of Normalized Difference Vegetation Index (NDVI) taking monthly precipitation and temperature data as inputs. The data generation is based on the simple relations between each attribute and climatic variables. In all cases neural networks performed better than RM and they have shown the ability to deal with the non-linear dynamic systems, complex in both time and space.

The models were trained for different locations in the catchment, different size of catchment and different rainstorm and duststorm events. The temporal distribution of rainstorm and duststorm was measured by coefficient of kurtosis. From the obtained result it was concluded

that the accuracy of the neural network models decrease with the increase in the size of catchment as well as with increase in the period between the two successive rainfall events. The best model result was obtained for the prediction of total suspended solid concentration.

### Rainfall runoff modelling

This type of application is one of the areas where ANN is applied most often. The traditional techniques to model rainfall-runoff process are mainly computationally demanding. ANN approach is used for replicating those models. In case of more complicated catchments the ANN models are used to simulate the rainfall-runoff process on the basis of measurement data. Minns and Hall (1996) investigated the use of multi-layer perceptron NN for rainfall-runoff modelling successfully.

Minns and Fuhrman (2000) are also studied the rainfall-runoff modelling in snow covered catchment on the basis of measurement data. Most of such process modelling do not take into account the influence of snowmelt water. The study focused on choosing appropriate input variables for a rainfall-runoff model for a river, where the majority of runoff is due to snowmelt processes. Study shows that the raw measured data cannot guarantee the good model performance. The study also demonstrates how the simple hydrological measurement can be manipulated and can improve a performance of ANN model remarkably without requiring additional measurements.

The RBF network was used for rainfall runoff modelling in drainage system (Mason *et al*, 1996). See also Mason and Price (1998). Comparatively good results were obtained by RBF network with radial centers fixed by a data clustering technique much rapidly than the error backpropagation network. The advantage of RBF network is much faster than the error backpropagation network and the traditional physically based modelling technique (average of 500 times).

See *et al* (1999) applied soft computing techniques including ANN and fuzzy inference model optimized by genetic algorithm for flood forecasting warning system. The applied ANN technique here is a hybrid of SOFM and MLP network. SOFM pre-classified the events into five groups prior to training with a set of 5 individual MLP networks. Training different networks for each event type is helpful for avoiding the errors caused by peak events.

Gautam (1998) applied regressive neural networks for modelling and forecasting the rainfall runoff relations in a Sieve river basin, Italy. Normal error backpropagation networks trained with data insertion is studied for the same case study as well. From the correlation analysis it was obtained that six-hour time lag and moving average value till 48 hours have high correlation with the runoff. The comparison of two different algorithms shows advantage of RNN above the training with data insertion for modelling rainfall streamflow relations.

The river flow prediction at certain sampling station of Huron river catchment in Germany by making use of measured discharges at three other sampling stations in the same catchment was addressed by Karunanithi *et al*, 1994. The cascade correlation networks used for the study and simulation result is compared with the *two-station power* model where model coefficients are estimated by least square regression. Two different neural network models were built: with current daily discharges at each three measuring stations (three inputs) and with 5 day non-overlapping average discharge for each station (total of 15 input nodes) as input variable. Different structures of neural networks are studied by trial and error in order to

get the best result. The result of the power model and neural network approach indicated that the neural network performs better than the power model in case of extreme flow situations. However, in low flow situations the two techniques perform with the same magnitude of accuracy. The neural network structure with input variables of 5-day discharge window without averaging performs better than the other structures.

Another application of ANN for filling in gaps in measurement data and for rainfall-runoff modelling was investigated by Kusumastuti (1999). MLP with error backpropagation learning algorithm is used for filling in missing data of one rain gauge station from daily rainfall data (antecedent, present and next day) of surrounding stations. The result obtained is not really satisfactory, however, on monthly basis the obtained result is more promising than daily data. Then Radial Basis Function (RBF) network is used for modelling streamflow on natural catchment in Indonesia and the result is compared with Chaos theory, Nearest neighbour technique and Marginal Storage Loss model. The inputs for neural network model to determine monthly average runoff of the catchment were monthly rainfall from surrounding stations. The result confirms the ability of neural network to identify the rainfall runoff relations, however, the chaos theory gives better result for prediction. Regression Nearest Neighbour method also performs better than the RBF network.

Yan (1999) studied the use of data driven modelling techniques for improving accuracy of flood forecasting. The following methods have been used on two natural catchments Bird Creek, USA and Yangtze River, China:

- chaos theory
- Ensemble Kalman filter in NAM model
- Mike 11 error prediction model in flood forecasting module
- ANN combination with NAM model

The Time Lag Recurrent Networks (TLRN) with precipitation, measured discharge and NAM simulated discharge as input variables are trained to predict the precipitation at two time step ahead ( $t+2$ ). The comparison of different techniques reveals that the chaos theory is the best for flood prediction. TLRN result for flood prediction is not yet acceptable especially for a complex river system like Yangtze river, but its combination with NAM model improves the accuracy significantly.

### Reservoir inflow prediction

Raman and Sunilkumar (1995) investigated the problem of modelling of monthly inflow to reservoir by ANN and statistical techniques. The study is based on the measured monthly inflow data of two reservoirs for a period of 14 years in Kerala, India. The input data for feedforward neural network model is organized in a way that the whole data set is divided into 12 monthly input data sets. No other input data is considered. The neural network model is built with 4 input nodes for two consecutive receding inflow for each reservoir and 2 output nodes, which are the third consecutive inflow for each reservoir. The autoregressive model for inflow performs well. However, in terms of the skewness analysis the ANN approach preserved the mean of the generated series better than the statistical technique.

### Stage-discharge relationship

A comparative study of conventional and the ANN techniques on discharge prediction from stage-discharge relationship has been carried out (Bhattacharya et al, 2000). The case study focused on a river in West Bengal, India, of which the measurement location has some odd



features to make a prediction more complicated. A backpropagation neural network model with 4 input variables was built for emulating the stage-discharge relationship. The measurement data of 6 years have been used for training and 3.5 years of data have been used for verification of the network. The comparison of the obtained result shows that the neural network model outperforms the traditional technique. In case of ANN model, 79.6% of validation data were within 5% of prediction error whereas for the traditional method it was only 57.6%.

### Cleanup of groundwater contamination

Cleanup of contaminated aquifer is a very complex and expensive problem. Usually pump-and-treat method is used for this purpose by installing and operating a set of extraction/injection wells for pumping out and treating the groundwater. The travel time of a contaminant is calculated on the basis of a so-called 'particle tracking' method and is a highly non-linear and convex function of pumping/injection rates and well locations. Therefore the global optimization techniques are used for determining the optimal pumping rates. The technique should be coupled with the running simulation model, which is a particle tracking models in this case. This kind of coupled model is computationally very demanding. The possibility to apply ANN for replicating the simulation model has been investigated (Maskey *et al*, 2000). The ANN model has been trained on the basis of the simulation data (pumping rates) to produce the optimal clean-up time or clean-up cost. The global optimization tool is then run parallel with the error backpropagation neural network model model, taking its output at each iteration to determine the optimal pumping strategy. The obtained result by ANN model shows a reasonable accuracy and the application of ANN reduces the required simulation time of the physically based model remarkably. The ANN must be trained on finer interval of decision variables in order to produce accurate result. The research can be extended further for a case with increased number of wells. Also it is advised to use ANN for finding the regions in the search space associated with higher probability of finding the global minimum in order to make the global optimization faster and more accurate.

#### *2.4.4 Coastal water systems*

### Controlling water level of drainage basin

Auto Regressive Neural Networks (ARNN) have been used for determining water level at the control location of Rijnland drainage basin in the north-east Netherlands (Werner and van den Boogaard, 1999). Excess water from the drainage area is discharged through the sluice gate to the North Sea. Thus water level must be maintained within a certain control band. Moreover, water is discharged by the sluice gate during the low tide, when the outside water level is 10 cm lower than the inside water level.

The system load as a consequence of precipitation is determined by rainfall-runoff model of Rijnland drainage basin. Hydrodynamic channel routing models are used to determine the water level at the control point. Using this load of the system, the control actions are determined at the control structure (opening of the sluice, pumping status). The system state obtained is coped with the actual state by data assimilation technique, which is not always robust and computationally demanding. On the other hand the prediction is based on a short-term rainfall forecast which is uncertain. Therefore ANN is used for rapid evaluation of the control procedure.

Water level is predicted by ARNN as a function of historical rainfall records (up to 96 hours), gate opening and outside water level. Inside water level and change of water level are determined as output. The network structure is defined with one hidden layer of 4 neurons. After simulation with different seasonal data, ANN results RMSE in order of 2-5 cm. The multiple simulation is performed in forecasting rainfall distribution and control action of sluice gate. If the water level exceeds minimum/maximum band then the control action is adjusted and the procedure is repeated until the probability of exceeding the band set is acceptably low. Due to rapid evaluation of the iterative procedure and the more reliable control strategy, this approach is concluded to be suitable for real-time operation.

### Current prediction in shallow coastal waters

A problem of finding an appropriate and reliable technique to predict current velocities in shallow coastal channel has been investigated by Wüst (Wüst, 1995). Several shallow channels at the Southern Bight, 25 km west of Amsterdam are dredged in order to make them accessible for the ships coming from the North Sea. The relatively strong cross channel current, which is important for navigational safety of ships, exceeds the safety standard of 50 cm/s at almost every tidal cycle. As a result the ship navigation is postponed up to one tidal cycle. The numerical models available are not yet used for operational forecast of current velocities. Neural network approach was successfully applied and is implemented for prediction of current velocities at the Hydro Meteo Centre, Hook of Holland.

The neural network prediction operates on-line and prediction for coming 24 hours is made 4 times a day. Frequency increases in case of stormy situations. The output is averaged from 4 separately trained network's outputs on different input variables. The training data set is developed with current, wind and water level measurements for a period of 9 months from the years 1988, 1989, 1992 and 1993 including several extreme hydrological situations. The current under investigation is nearly perpendicular to the channel axis. Therefore, special importance is given to the choice of the wind and current measurement point. Input variables for training neural networks include current velocity, water level, average wind velocity for 2 preceding tidal cycle and the deviation vector for wind over periods of increasing length going back in time (1,4,8 and 16 hours back).

The network's performance can be described through Root Mean Square Error (RMSE) which was, on an average, 13.4 cm/s. Under the astronomical current conditions it gets a value of 8 cm/s. However, the performance is poor during the high winds and during the ebb tide. Neural network model predicts the current velocities from the wind and water level measurements successfully. It should be noted that during extreme hydrological situations, which is not well represented in the training data set, would result in poor performance of the model.

### Water tide prediction

Prediction of tidal level at the big harbours is an important task not only for storm surge situations but for daily management tasks as well. The prediction system with highly complex modelling programs involves a huge amount of data and it requires a personal experience (Breitschdel *et al*, 1998). Therefore, one way to incorporate the domain knowledge into the numerical modelling technique can be the use of neural network and other artificial intelligence techniques. In this study, neural network is used first to find an association between data sets at different locations along the Dutch coast in order to explain the reason

behind the repeated appearance of faulty signal. Secondly, the neural network is applied for predicting water level and wind components at certain locations. The accuracy of the result of neural networks for above purposes was satisfactory.

The authors stressed the importance of developing the integrated classification and prediction support tool for operational management for storm surge department. Neural networks learn the similar storm situations characteristics on the basis of past information and produce the expert an indication how the storm situation may develop and what water level it should cause etc. The integrated tool should have common user interface with three different modules or tools:

- model evaluation tool, which is numerical model part
- neural network tool, which also will be manipulated on the existing data base
- decision support tool, which will be knowledge based and rules should be updated after each event for future use

The advantage of such integrated tool is that it connects the numerical modelling technique with the AI technique to include the advantages of both. The tool integrates itself in a way of updating the knowledge base at each time the new storm situations occur and this serves as the principal advantage of this tool.

#### Determining the erosion of field sediment

The sediment erosion of the groyne field is one of the important issues in large cargo handling ports like Rotterdam. The probable relation between the characteristics of navigating vessels and the quantity of erosion in the groyne field has been investigated by using error backpropagation ANN with two hidden layers (Schulze and Salverda, 1999). The study is based on a total of 9 measured and derived characteristics such as water depth, wetted cross section of the river, speed, length, width, corrected speed and wetted cross section of the ship, the distance of the vessel from the groyne and duration of navigation. The erosion rate is obtained as an output of the network. It has been concluded that ANN's quantification of the erosion of groyne field from the vessel characteristics is adequate enough.

#### Prediction of error in the solution of hydrodynamic model

The choice of appropriate spatial and temporal resolution is one of the key steps in ensuring accuracy in mathematical modeling. It is impossible to find an ideal resolution as long as the flow variables vary spatially throughout the simulation period. Some advanced commercial modeling packages deal with this problem by altering the time step depending on the parameters such as celerity, convergence, etc. This solution is usually computationally demanding. Abebe et al (2000) have shown the possibility of using ANN to reproduce the discrepancy in the water level prediction as a result of inappropriate fixed computational time steps. The study considered a 1D model of an estuary solved using the full de Saint Venant equation. The Radial Basis Function network was trained to predict error between the solutions with the fixed and the variable time step intended to keep the Courant number close to unity (assumed to be the exact solution). The input variables are the depth and discharge at the current time step and outputs are the error at the next time step. After being trained the network is used parallel with the physically based model for the rest of the computation. The neural network can predict the discrepancy of the solution with a high accuracy (correlation

coefficient of  $r^2=0.971$ ). The result shows that the ANN can be used as an integrated complementary component to mathematical models.

## 2.5 Practical issues of using NN for engineering applications

### 2.5.1 Introduction

Where processes to be modelled are complex enough to be described mathematically, neural networks are considered to outperform the conventional, deterministic models most of the time. However, one should be aware of the applicability of neural networks to a specific problem and the basic conditions for getting the best performance out of it. In many cases neural networks for research are used 'blindly' by choosing all the possible input variables and without considering much of the possibilities to maximize the performance.

The purpose of this section is to provide the reader some practical information of taking the maximum advantage of the neural network models. Moreover, it should be noted that it was not aimed to give a complete recipe of using neural networks, the reader should get from this section rather a general view of what are the most important issues to be taken care of, in order to work successfully with neural networks. Mainly the section is based on Swingler (1996) and partly on Kolb (1999) and the reader may refer them for more details.

### 2.5.2 Analysing the problem

In general, neural networks are suitable for problems where the underlying process is not known in detail and the solution can be learned from the input-output data set. Nevertheless, the following points have to be stressed:

1. It has to be made sure that the problem is difficult to be solved by conventional method and neural network can be used as a good alternative.
2. If there are logical nonchaotic relationships or structural properties that similar initial configurations indicate mapping to the similar solutions, one can expect a generalization by neural network. It simply means, the same input should always result in the same output.
3. If the data set to train the network is impossible to be represented or coded numerically, the problem cannot be solved by a neural network approach
4. Non-linearity and the change of variables in time are possible to be dealt with neural networks.

### 2.5.3 Data preparation and analysis

This is one of the most important stages of neural network application because the accuracy of solution for most of the networks depend on the quality and quantity of training data set. Although neural networks can accept a wide range of inputs, they work with data of certain format encoded numerically. There are two main issues in data preparation:

- The number of variables to be used, which determines the dimensionality
- Explicitness or data resolution and in what extent and amount the data has to be presented to the network

To avoid of analysis of large amount of data, a sample data set may be used by choosing it randomly from the complete database. For input and as well for output variables the data must be analysed and prepared with the following sequence, which is sometimes called as data pre-processing:

1. Determining the data type (discrete or continuous)
2. Data generation. The data to train the network can be generated by measurement, by simulation of relevant models or by derivation of virtual examples by introducing noise into the existing data set. Also it is good if the data set evenly covers the input data subspace. In other words the data has to be normally distributed.
3. Calculations of simple statistics such as mean, standard deviation for continuous data and the number of different events for discrete data.
4. Removal of outliers. By outlier we mean the data points lay outside of two standard deviations from the mean. Two standard deviations cover 95% of normally distributed data. If such data example exist, those are preferably to be removed, unless those are significantly important for the given problem. For some of the dynamic systems (chaotic) those outliers are important.
5. Quality and quantity check. What amount of data has to be collected is mainly decided by the network size (number of variables), required data resolution etc. Concerning the network size it is advisable to collect training data set of equal number to  $(1/\text{target error}) \times \text{number of weights}$ . Also as a quality check statistical tests can be carried out in order to make sure the corresponding data set contains a required information.
6. Dimensionality reduction. Large number of input variables increases the training time considerably. It is advisable to reduce the number of input variables, which are the most important and best representing the output variable while maintaining the correct level of network complexity. The covariance or correlation between the variables can help to decide which variable is the most useful.
7. Data scaling has to be done when data set has too different order of magnitudes. It is also advisable to have all the input data within the same range of scaling.
8. Data encoding has to be done in the end of data preprocessing in case of necessity. Categorical data must always be encoded.

#### 2.5.4 Model selection and building

Because of its accuracy and fault tolerance capability error backpropagation network is the mostly used type of neural network. However, there are different types of learning algorithms that are quite suitable for specific problems. For time dependencies the recurrent neural networks and for classification the Kohonen networks are well suited. Also for feature extraction and classification purpose Principal Component Neural Networks (PCNN) are applicable etc.

For classification almost all types of networks are applicable. Supervised learning algorithms classify the data into predefined groups while unsupervised neural networks have self-organising features to find unknown regularities and patterns in the presented input data and they are capable of finding the hidden features. Kohonen networks map the input vectors into one or two dimensional topology preserving output layer, Hopfield network find the nearest match among the stored patterns. Learning Vector Quantization classifiers use non-linearly separable vectors and the Adaptive Resonance Theory defines its own classification groups.

In case of too many input and output variables, the training of the network become computationally demanding. Therefore, one way to solve this sort of problem is to divide the problem into several small sub-problems that can be solved separately by the network. There is no specific rule for building a network, however, some practical hints on this aspect is listed below.

### *Network structure approximation*

#### *Multi-layer Perceptron*

- The training examples should be at least equal to  $1/\mathbf{e}$ , where  $\mathbf{e}$  is a target value for error
- The maximum number of hidden units should be guided by the formula  $h \leq 2i+1$ , where  $h$  and  $i$  are number of hidden and input units respectively
- Number of weights can be related to the number of training patterns  $w = i \log_2 p$ , where  $p$  and  $w$  are number of training patterns (exemplars) and number of weights respectively
- For feature extraction, number of hidden nodes should be less than the number of input variables
- For classification, the number of hidden units is increased with the number of expected classes
- Number of hidden layers should be as less as possible and usually one or two layers are used in most of the published applications. It was shown that any function could be approximated by at most 4 hidden layers (Swingler, 1996).
- It is suggested that the activation function in the specific neuron has to be chosen as non-linear for non-linear process model. In case of more than one hidden layers, the activation function for one of the layers has to be linear, in order to discard the linear components that may be existing in non-linear models.

#### *Self Organising Feature Maps*

SOFM consists of the input layer and the output map. Concerning the size of the network following rule is mainly suggested:

$$2N_{class} < N_{units} \ll N_{pattern}$$

$N_{class}$  - the number of expected class or cluster (user must have some primary expectation)

$N_{units}$  - Number of processing units in the output layer

$N_{pattern}$  - Number of input pattern

#### *2.5.5 Training and testing the network*

Training is the learning process of neural networks. Training stage can be started when the network is designed, data sets are collected and encoded. After the training is fulfilled, the testing phase starts.

*Defining the topology* Training the network has to be started by defining the topology of the neural network. The best topology is found by adjusting the parameters by trial and error, therefore it is better to start with a small network which learns fast and is easy to change the parameters. Initial weights are also defined by trial and error method. When the appropriate network topology is defined, it is possible to speed up or slow down the process by changing the learning rate and make fine-tuning.

*Stopping criterion* There is no specific rule signifying when to stop the training and the stopping criteria are different for different type of the network. In general, the stopping criterion can be the minimum value for a learning rate or if applicable, it is also advisable to use target value of training error and also an option of cross validation, which allows you to know whether the error in verification phase starts to increase.

*Testing* Once the network is considered to be sufficiently trained, the network needs to be tested under realistic circumstances. However testing is not necessarily applicable for every type of neural networks. The final integration or implementation of the neural network has to be delayed till sufficient confidence is achieved, so that the network can work by all means and without damaging the system in case of its failure.

### 2.5.6 Output and error analysis

Errors do not always mean the network parameters are chosen wrong. If the network is built and organised systematically then the reason for large error can be found by changing few parameters by small amount between the two configurations. But if the problem cannot be found, the reason is not in the initial configurations. Sometimes errors or unsuccessful results can not be simply termed as errors as they might be caused by uncertainty.

Error criteria can be a maximum net output error, which is the difference between the net output and the target output. Average error and moreover, the total distribution of error are good performance criteria. If network makes error in some cases and not in others, it means the data balance is proper. In order to prevent the accumulation of rounding error, training the network with too large or too small numbers should be avoided.

For noisy and overlapping classification problems it is impossible to get zero error level. Function mapping is more difficult than the classification problem and the different parts of the input space can give a different degree of error. Certain parts of a data set can represent non-linearity function that can be learned easily. Also there can be more variance in training data set than in the other parts. In short, the inadequate generalization usually is caused by too complex non-linearity of the function, too high variance in data set.

There are different ways to evaluate and analyse the network output error. One easy way to analyze the output error is to calculate simple statistics of the network output such as the correlation coefficient between the net output and the target output. One more advanced method to evaluate and minimize the error is finding the structure of error, which could determine the areas with different level of predictability. Even the structure can be the function describing the distribution of error in the input space. This goal can be achieved by training the second neural network to learn and predict the error of the original model after its being trained at certain level (for details see Swingler, 1996).

### 2.5.7 Implementation of a neural network based project

This is the step to build the real product from the neural network prototype. Implementing the neural network is the part of the software we need. It includes special requirements such as time or space restrictions, porting the neural network solution to an application environment and interface development etc. Most of the time the neural network project can be easier than to the rule-based approach as the domain specific knowledge is not much required for neural

networks. In terms of risk involved in neural network project, the main risk would be the non-presence of the information necessary for the problem in the data set available.

There are three general steps in implementing a neural network based project, each of which consists of small substeps.

- Project planning stage ( task definition, feasibility study, input/output specification, defining data requirement, data coding)
- Network development stage (data collection and validation, data encoding/recoding, network design and training, network testing and error analysis, implementation)
- Documentation stage (defining data source and conditions, defining the coding method, architecture, parameter setting, the number of training epochs, defining the conditions, reporting the final results).



## Chapter 3 Fuzzy logic approach and applications

### 3.1 Introduction

The origin of the fuzzy logic approach dates back to 1965 since Lotfi Zadeh's introduction of the fuzzy-set theory and its applications. Since then the fuzzy logic concept has found a very wide range of applications especially in the industrial systems control those are very complex, bear uncertainties and cannot be modelled precisely, even with simplified assumptions and approximations (Hirota, 1993).

This chapter introduces the basics of the fuzzy logic approach and the Fuzzy Adaptive Systems (FAS) followed by a review of their applications in the water management field.

### 3.2 Basic concept of fuzzy logic approach

The fuzzy logic systems can be seen as *structured numerical estimators* (Lin 1994). In the fuzzy logic approach the Boolean logic is extended to handle the concept of *partial truth* which implies that the truth takes a value between a completely true value and a completely false value. For example, the partial truth can have values in linguistic variables like not very truth, more or less false etc. To accomplish this idea the notion of the fuzzy *sets* has to be introduced, which is the collection of the objects that might belong to the set to a degree, taking any values between 0 (full non-belongingness) and 1 (full belongingness), instead of taking a crisp value (0 or 1).

As mentioned above, the fuzzy logic approach is particularly a preferable tool for dealing with problems with uncertainties and imprecise information. However, the distinction should be made between the uncertainty due to randomness and the uncertainty due to the imprecision. Imprecision is an absence of a sharp boundary and exactness in the information, while the randomness is about the occurrence of the event itself.

The indication of intensity of belongingness is expressed by the *membership* function, assigning each element a number from the unit interval  $[0, 1]$ . Let  $X$  be a universal set then  $A$  is called the subset of  $X$  if  $A$  is a set of ordered pairs.

$$A = \{(x, m_A(x); x \in X, m_A(x) \in [0,1])\} \quad (3.1)$$

Where the function  $m_A$  is the membership function of  $A$ .  $m_A(x)$  is the grade of the membership of  $x$  in  $A$ . For example, if the set of young persons is fuzzy then a person with 25 years of age can be young with a truth value of  $m_A(x)=0.9$  etc. In this way the crisp numbers are fuzzified. The shape of membership functions can be of different types, such as triangular, trapezoidal, bell-shaped etc.

The basic operations of mathematical sets like complement, intersection and union are also performed on fuzzy sets. The *Fuzzy rules* consist of arguments coupled by logical operators and are verbally formulated as: *IF the condition is fulfilled THEN the consequence has to be true*. The logical expressions are usually formulated by logical operators *AND*, *OR*, *NOT* and *XOR*. The truth value corresponding to the fulfilment rule conditions for a given premise is

called the *degree of fulfilment* (DOF). The most commonly used methods to determine the DOF are *product* and *min-max* inferences. Then the rules will be responded in different combinations. These combinations are *minimum*, *maximum* and *additive* combinations. As an example let us consider the minimum rule response. For details the readers may refer to Bardossy & Duckstein, 1995.

The minimum combination method tries to find a rule response, which, at least, remains to a certain level in DOF with all the applicable rules. Considering the rule responses  $B_1=(0,2,4)$  with DOF  $n_1=0.4$  and  $B_2=(3,4,5)$  with DOF  $n_2=0.5$ , the minimum combination of responses ( $B_i, n_i$ ) is the fuzzy set  $B$  with the membership function:

$$m_B(x) = \min_{i>0} n_i m_{B_i}(x) \tag{3.2}$$

where  $m_{B_i}(x)$  is a membership function of  $x$  in fuzzy set  $B$  for rule  $i$ .

$B_1$  and  $B_2$  have intersection  $[3,4]$ . The equation taking the minimum of two membership functions on  $[3,4]$  will be as follows:

$$\min\left(0.4 \frac{4-x}{2}, 0.5(x-3)\right) \tag{3.3}$$

which results in:

$$0.4 \frac{4-x}{2} = 0.5(x-3) \tag{3.4}$$

Subsequently the value of  $x=23/7$ . Thus the minimum of membership function is:

$$m_B(x) = \begin{cases} x-3, & \text{if } 3 \leq x \leq \frac{10}{3} \\ \frac{4-x}{2}, & \text{if } \frac{10}{3} \leq x \leq 4 \end{cases} \tag{3.5}$$

The graphical representation of this combination is shown in Figure 3.1. The thick line is used for the minimum combination and the thin lines for the individual rule consequences.

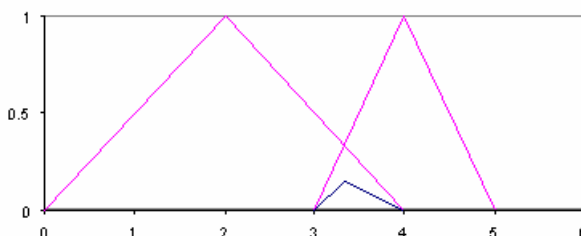


Figure 3.1: Membership functions for consequence  $B$  using minimum combination (Bardossy and Duckstein, 1995)

A collection of fuzzy rules can be used to form a control algorithm. The basic structure of the fuzzy rule based system (FRBS) involves four principal components (Figure 3.2).

- *fuzzification interface*, where the values of the inputs are measured, fuzzified and the input range is mapped into the suitable universe of discourse .
- *Knowledge-base*, which involves a numeric ‘database’ section and a fuzzy (linguistic) rule-base section.
- *Fuzzy inference* mechanism or engine, which constitutes the core of the FLC, involves the decision making logic (fuzzy reasoning such as product, max-min composition etc)
- *Defuzzification interface*, which maps the range of output variables into the corresponding universe of discourse and defuzzifies the results of fuzzy inference mechanism. The defuzzification methods can be *maximum* of the rule consequence, *mean* and *median* methods (Bardossy and Duckstein, 1995).

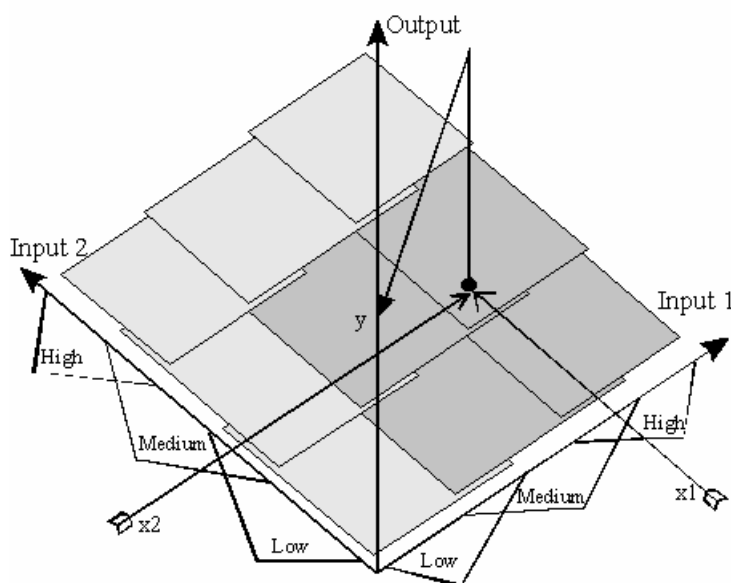


Figure 3.2 Principle of FRBS response (adopted from Abebe et al, 1999)

An example of a simple FRBS with two inputs and one output and the principle of fuzzy rule base response is presented in the Figure 3.2 (Abebe, 1999). Input  $x_1$  belongs to the *high* and *medium* and input  $x_2$  belongs to the *low* and *medium* membership functions at the same time with a different degree due to overlapping. Two inputs activate four out of nine rules constructed from three input membership functions. The DOF determined by the fuzzy inference results the defuzzified numerical output  $y$ .

Another example can be a reservoir with a possible water release into a river and the three associated so-called linguistic variables: *precipitation*, *water-level* and *reservoir-release*. The values of linguistic variables are not exact real ones, but rather fuzzy, like *LOW*, *MEDIUM*, *HIGH*, etc. Figure 3.3 presents the variable water-level. Based on the human experience or data sets processed one can construct so-called *fuzzy rule* as:

“IF *precipitation* = *HIGH* AND *water-level* = *MEDIUM*  
THEN *water-release* = *MEDIUM*”

### 3.3 Fuzzy Adaptive systems

Getting several fuzzy rules from an expert’s knowledge is not too complicated for a simple case. In a complex system, which is usually the case, the scope of construction of the rule-based system is limited (manipulation and verbalisation of variable). Therefore, the

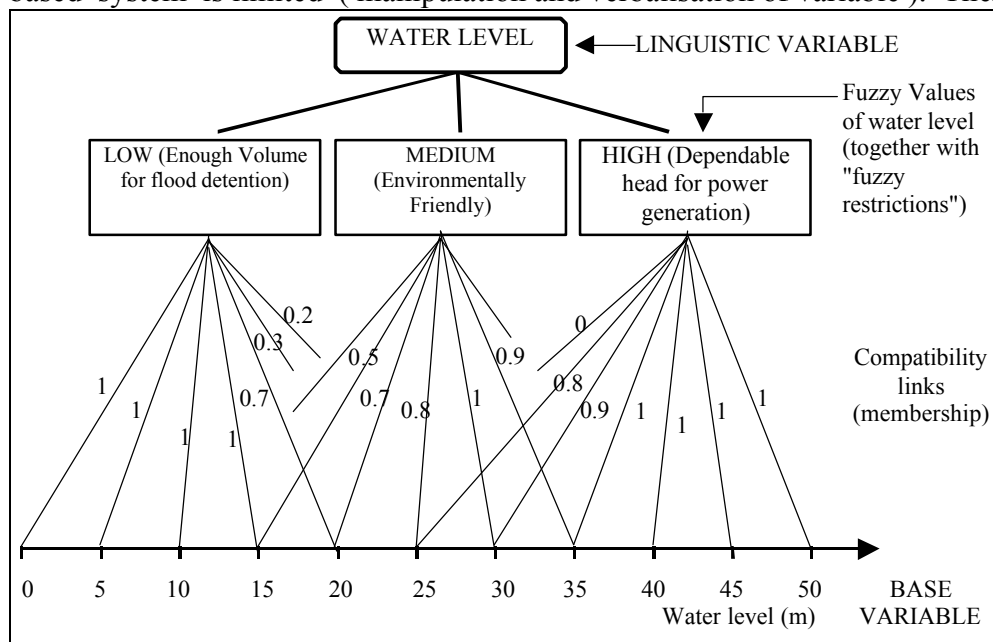


Figure 3.3: Water level in a reservoir expressed through linguistic variables

possibility of inducing and learning the rules from data has been investigated and implemented successfully and these systems are called Fuzzy Adaptive Systems (FAS). On the basis of the user defined input membership functions and input-output sets, FAS can determine the output membership functions and defuzzified outputs.

There are different methods to derive the rules directly from a data set: *counting* algorithm, *weighted counting* algorithm and *least squares* algorithm. The principle of counting and weighted counting algorithm is nearly the same, only in the case of weighted counting algorithm DOF is used for determining the rule response. A brief description for the counting and least squares algorithms is given below (for details see Bardossy & Duckstein, 1995).

#### Weighted counting algorithm

For a given relevant variables the fuzzy rule based system has to deliver the response close to the observed one. The set of training example  $T$  has to consist of the input  $a$  and output  $b$  and is written as follows:

$$T = \{(a_1(s), \dots, a_k(s), b(s)); s = 1..S\} \tag{3.6}$$

If all the variables and the responses are continuous then the rules can be constructed by defining the fuzzy set that supports the fuzzy numbers  $A_{i,k}$  and identifying the corresponding responses. The following algorithm is used.

1. Define the membership function of the premises.  $A_{i,k}$  is assumed to be a fuzzy number  $(\mathbf{a}_{i,k}^-, \mathbf{a}_{i,k}^l, \mathbf{a}_{i,k}^+)$  where  $\mathbf{a}_{i,k}^l$  is the mean of all possible  $a_k(s)$  values which fulfil at least partially the  $i^{\text{th}}$  rule :

$$\mathbf{a}_{i,k}^l = \frac{1}{N} \sum_{s \in R_i} a_k(s) \quad (3.7)$$

where  $N$  is the number of elements in  $R_i$ .  $R_i$  is the set of all those premise value vectors that fulfil at least partially the  $i^{\text{th}}$  rule and it forms the subset of the training set  $T$ .

$$R_i = \{(a_1(s), \dots, a_k(s), b(s)) \in T; a_k(s) \in (\mathbf{a}_{i,k}^-, \mathbf{a}_{i,k}^+); k = 1, \dots, K\} \quad (3.8)$$

2. Calculate the Degree of Fulfilment (DOF)  $v_i(s)$  for each premise vector  $(a_1(s), \dots, a_k(s))$  corresponding to the training set  $T$  and each rule  $i$  of which the premises were defined in the previous step.

3. Select a number  $\mathbf{e} > 0$  such that only responses with DOF of at least equal to  $\mathbf{e}$  will be considered in the construction of rule response. And the corresponding response is also a fuzzy number  $(\mathbf{b}_i^-, \mathbf{b}_i^l, \mathbf{b}_i^+)$  where  $\mathbf{b}_i^-$  is the minimal,  $\mathbf{b}_i^l$  is a weighted mean and  $\mathbf{b}_i^+$  is a maximal value with a degree of fulfilment of at least  $\mathbf{e}$ .

$$\mathbf{b}_i^- = \min_{v_i(s) > \mathbf{e}} b(s) \quad (3.9)$$

$$\mathbf{b}_i^l = \frac{\sum_{v_i(s) > \mathbf{e}} v_i(s) b(s)}{\sum_{v_i(s) > \mathbf{e}} v_i(s)} \quad (3.10)$$

$$\mathbf{b}_i^+ = \max_{v_i(s) > \mathbf{e}} b(s) \quad (3.11)$$

The value of  $\mathbf{e}$  has to be selected so that sufficient number of elements of the training set is considered for each rule.

### Least squares algorithm

This algorithm is based on the traditional function fitting method of least squares. The algorithm has certain restrictions; for determining the rule response the normed weighted sum combination method and for defuzzification the mean method has to be used. The algorithm can be formulated as follows:

1. Define the membership function of the premises using the same way as in the weighted counting algorithm.
2. Calculate the rule response by minimising the sum of squared error of rule system  $R$  resulting from using the rules. The function to minimise should be as follows:

$$\sum_s [R(a_1(s), \dots, a_k(s)) - b(s)]^2 = \sum_s \left( \frac{\sum_{i=1}^I \mathbf{n}_i(s) M(B_i)}{\sum_{i=1}^I \mathbf{n}_i(s)} - b(s) \right)^2 \quad (3.12)$$

where,  $M$  is a fuzzy mean of rule response  $B_i$ . As long as the rule response for normed weighted sum combination method:

$$R(a_1(s), \dots, a_k(s)) = \frac{\sum_{i=1}^I \mathbf{n}_i(s) M(B_i)}{\sum_{i=1}^I \mathbf{n}_i(s)} \tag{3.13}$$

The final goal of optimization should be fulfilled when the derivative with respect to the unknown  $M(B_j)$  becomes 0. Differentiation for every index  $j$ :

$$\sum_s \left( \frac{\sum_{i=1}^I \mathbf{n}_i(s) M(B_i)}{\sum_{i=1}^I \mathbf{n}_i(s)} - b(s) \right) \frac{\mathbf{n}_j(s)}{\sum_{i=1}^I \mathbf{n}_i(s)} = 0 \tag{3.14}$$

Rearrangement of the equation will give the system of  $I$  linear equations for  $I$  unknowns  $M(B_1) \dots M(B_I)$ .

$$\sum_s \frac{\sum_{i=1}^I \mathbf{n}_i(s) \mathbf{n}_j(s) M(B_i)}{(\sum_{i=1}^I \mathbf{n}_i(s))^2} = \sum_s \frac{\mathbf{n}_j(s) b(s)}{\sum_{i=1}^I \mathbf{n}_i(s)} \tag{3.15}$$

3. The final rule system will be as:

$$\text{If } A_{i,1} \text{ AND } \dots \text{ AND } A_{k,1} \text{ then } M(B_i) \tag{3.16}$$

The disadvantage of this algorithm can result in the fuzzy mean, which is not reasonable for the individual rule response to that particular rule.

### 3.4 Fuzzy logic control

The fuzzy logic control (FLC) has been used much more extensively in practical applications than the neural network control. There are different types of FLC distinguished:

- *self-organizing* FLC which measures its own performance and modify the control rules according to the error,
- *supervising the conventional controller*, where conventional PID controller is supervised by a fuzzy logic supervisor (Tzafestas, 1997).

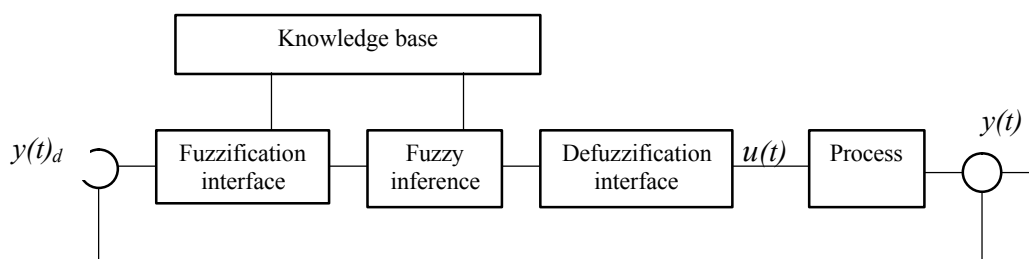


Figure 3.4: Fuzzy logic control scheme

Mainly, the FLC is designed for the SISO (Single-Input-Single-Output) and the MISO (Multiple-Input-Single-Output) control problems. A basic structure of a fuzzy logic control is shown in Figure 3.4.

The advantages of a fuzzy logic control (FLC) can be summarized as follows:

- they can imitate the control actions of human operators through the description of the system behaviour using linguistic expressions
- they are inherently non-linear and therefore, able to perform the control actions that are not possible purely with a traditional linear control

### 3.5 Application of fuzzy logic approaches

#### 3.5.1 Sewerage systems

##### Controlling activated sludge process

Determining the optimal dosages of chemicals in wastewater treatment plant (activated sludge) process has been studied using a fuzzy logic controller built by the Fuzzy toolbox MATLAB with a combination of traditional techniques (Kalker *et al.*, 1999). Most of the aeration process is controlled through a direct DO (dissolved oxygen) controller. The PID controller switches on the aeration on the threshold value of ammonium and nitrate concentration. The performance of this type of controller is not always reliable and optimal, because the activated sludge is a highly non-linear and dynamic process. The treatment of water with heavy nitrogen concentration (up to 0.021 kg N/kg MLSS.day) with internal water re-circulation was considered in the study. The fuzzy logic controller was used to determine the optimal aeration rate in order to get the desired concentration of ammonium and nitrate from the system.

Aeration zone of the nitrogen removal process consists of the de-nitrification, the intermittent and the nitrification zones. The FLC determines the ratio  $d$  between the aerated period and the total cycle length  $T$  (total time required of aerated and unaerated periods). The scheme is shown in Figure 3.5. The input to the FLC is the difference of the desired and the current ammonium concentration  $\Delta NH_4-N$  and the change of ammonium concentration in time  $dNH_4-N$ . The efficiency of the controller is evaluated by considering the ammonium and nitrate concentration of treated water and the energy consumption, which is expressed in airflow used in the intermittent zone.

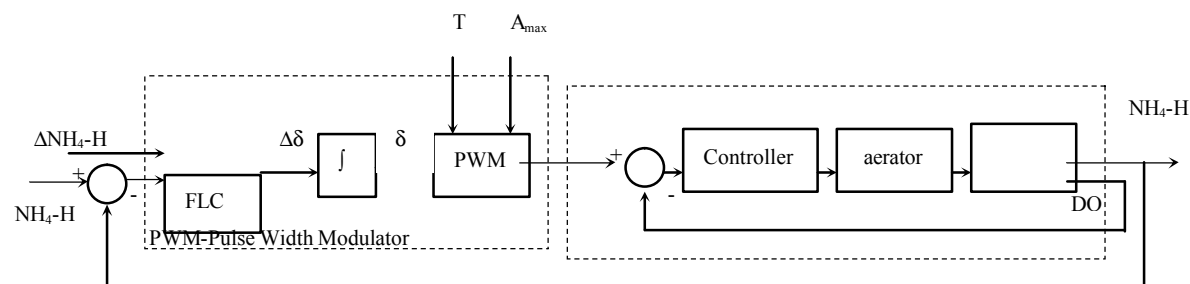


Figure 3.5 Block diagram of FLC

A fuzzy rule base was constructed with 9 fuzzy linguistic rules with input membership functions having triangular and parallelogram shapes. The output  $d$  is determined by a crisp membership function. The controller was tested for the dry and wet (rainy) periods. The result of the simulation is compared with the relay controller and the fixed ratio controller. Relay controller is where aeration is started with the exceeded ammonium concentration and stopped with the ammonium concentration less than the setpoint. The fixed ratio controller

uses the ratio between the fixed total cycle length and a fixed ratio between the aerated and un-aerated periods.

The obtained result shows that the ratio-controller performs better than the other types of controllers. Energy consumption for the ratio-controller is reduced by 10-23% compared to the feedback controller and the fixed ratio controller, which is expressed by the oxygen flow rate. The effluent water quality (of treated water) is improved by 0.6 mg/l than the value under the feedback controller and the variation of the effluent water quality is less than that under the other controllers, especially for the wet period where the variation is high. This type of controller can be successfully utilised for water purification problems. However, the direct fuzzy controller can be improved further by considering the multivariable aeration control such as choosing different setpoints.

### Suspended solid control of activated sludge process

Controlling a suspended solid concentration of the effluent water is a crucial issue in the activated sludge process, which is highly non-linear and dynamic in nature. A fixed ratio controller that keeps the ratio between the Biological Oxygen Demand (BOD) and the suspended solid concentration becomes unreliable especially during the peak load. Controlling the suspended solid concentration by implementing an on-line fuzzy logic controller is investigated by Tsai *et al* (1996). The case study considered the municipal wastewater treatment plant in the city of Taipei, of which the inflow rate, organic load and Chemical Oxygen Demand (COD) concentration vary immensely with the load period during a day.

In order to improve the control over suspended solid concentration, the fuzzy controller based on Newton's method is built on the basis of the measured input and output data. The input, control and output variables of the fuzzy logic controller are the inflow rate ( $Q_{in}$ ), the return flow rate ( $Q_r$ ) and the effluent suspended solid concentration ( $SS_{eff}$ ) respectively. In total, for each variable 8, 10 and 10 fuzzy clusters are considered. The return flow maintains the normal chemical reaction in the aeration tank, so that during the low load period the part of the sludge is pumped back to the aeration tank. But there should be no return flow when the inflow rate is higher than the daily average inflow rate. The fuzzy relation based on the Newton's method is written as follows:

$$SS_{eff}(t) = Q_r(t - 0.5) \Theta Q_{in}(t - 8) \Theta R \quad (3.17)$$

where  $\Theta$  denotes the min-max operation in the fuzzy set theory and  $R$  denotes the fuzzy relational set. The time delay factors introduced for  $Q_r$  and  $Q_{in}$  are 0.5 and 8.0 hours respectively. By using the existing training data the effluent concentration can be predicted with high accuracy. At the same time the desired value to be obtained by choosing a correct value for the return flow rate ( $Q_r$ ). The optimum control strategy derived from the fuzzy model was (Equation 3.17) tested on the automatic pilot scale plant. The obtained result clearly shows that the strategy derived by the fuzzy controller is able to reduce the suspended solid concentration much more than that with a fixed ratio controller especially during the peak loads.

Similarly the fuzzy adaptive model was built for characteristics of the activated sludge process (Scheffer, 1999). The model was also able to predict four parameters one time step ahead, although the accuracy of the prediction was not satisfactory.



### Combined Sewer Overflow controlling

Hou and Ricker (1992) established a fuzzy logic control for minimising Combined Sewer Overflows (CSO) in a three-reservoir system. During the stormy events, collecting and keeping the runoff flows in the reservoirs is one of the ways of preventing sewer overflows. However, the load of such system is variable in time and space and there are too many variables for dynamic response of the system. Therefore, the low cost control for the system to substitute the expert knowledge is desirable. The system in the study represents the network of two reservoirs with the same outflow rate connected to the third reservoir together. Simplified diagram can be seen in Figure 3.6.

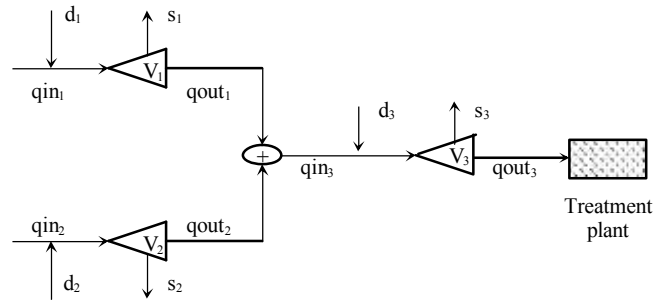


Figure 3.6: Diagram of three reservoir system

(Note:  $q_{in}$ - inflow,  $q_{out}$ - outflow of the reservoir,  $V$ - volume,  $s$ - overflow,  $d$ - sum of all lateral flows)

The fuzzy logic control was established to minimise the overflow from these reservoirs making full use of these reservoirs during the significant storm events. A special variable is used for control, which is a relative fullness within the reservoirs, in order to keep the storm runoff equally distributed in the area. The relative fullness within the system is calculated as follows:

$$df_{1-2} = \frac{V_1}{C_1} - \frac{V_2}{C_2} \quad (3.18)$$

$$df_{1,2-3} = \frac{V_3}{C_3} - \max\left(\frac{V_1}{C_1}, \frac{V_2}{C_2}\right) \quad (3.19)$$

Where,  $C_i$  - maximum capacity of the reservoir  $i$

$df_{1-2}$  - difference between the fullness of reservoir 1 and reservoir 2

$df_{1,2-3}$  - difference between the fullness of reservoir 1 and that of the higher occupied one in reservoir 1 and 2

The fuzzy logic controller takes these two inputs and uses 5 input membership functions to produce two outputs, which are the outputs of the first two reservoirs by 6 output membership functions. For input and output membership functions trapezoidal functions were used. 25 fuzzy rules were constructed, Mamdani rule was used for implication and centre of gravity method was applied for defuzzification. Example of constructed rules is as follows:

If( $df_{1-2}$  is PS and  $df_{1,2-3}$  is NL), then ( $q_{out1}$  is L+ and  $q_{out2}$  is M+)

The simulation was carried out between two successive events. However, for heavy storms the controller lacks in the predictive capability, because it was designed for the instantaneous response only. This causes overflow. If weather prediction module is introduced and the control actions in certain control horizon are optimised, the scheme would be more successful.

#### Determining water treatment dosage

The general purpose FLC incorporated into one-chip microprocessors and programmable controllers, was used to develop the so called FRUITAX system that has been applied for the water treatment and rain water pumping processes (Hirota, 1993). FRUITAX performs basically max-product, centre of gravity calculation. The result of chemical injection rate control in water treatment process was compared with the expert's decision, and the rain water pumping process control was compared with the conventional PID controller. The application of the system into these fields of process control shows the reliability of the system itself and the study has widened the range of its application.

#### Pollutant load estimation

The need for an accurate model to estimate water quality of runoff from urban areas or some critical sites such as mining and construction sites, are being increased recently. The necessity was spurred by the inaccuracy of conventional methods, regression or build-up-wash-up models and the efforts required for their calibration. The deterministic models do not consider the interactive processes such as pollutant transport, in-pipe sedimentation, interaction and re-suspension. These processes are indeed less understood. Baffaut and Chameau (1990) used the fuzzy set concept as an alternative way for the pollutant estimation and prediction. They modified the existing urban wash-up model (called SWMM) by including the fuzzy sets to estimate the uncertainty and pollutant load, with its associated calibration method and applied into two different watersheds. The obtained result shows an improvement on the accuracy of a model, where the original model with correctly determined parameters estimated the pollutant load with an error as high as 30%.

De-nitrification process has been controlled by the fuzzy logic controller (Aoi *et al*, 1992). Usually the de-nitrification process is controlled by the indirect components such as DO and ORP, whereas the authors included the ammonium as an additional component to monitor. Based on these parameters, direct ammonium control using a fuzzy inference has been built. The controller uses the above mentioned measured components and produces the air flow rate. The controller performed a quick response and high nitrogen removal rate with more stability and easy maintenance.

#### Sequencing wastewater treatment process

The fuzzy expert system was applied for sequencing the treatment processes in the treatment plant and was reported by Yang and Kao (1996). Wastewater treatment is a sequence of many different unit processes depending on the type of contaminant and their concentrations. The sequence of the processes is called a treatment train and designing the appropriate treatment train for a given inflow is a three-stage problem. The study concentrated on the first stage, which is used for making the list of possible treatment trains.

The objective of this technique is to choose the candidate techniques to satisfy the user's preference. Preference of the user is defined by the performance efficiency and construction cost of techniques. The system first takes the list and identifies the performance efficiency and the construction cost for each of the techniques. The user can determine the preference in terms of the cost and the efficiency, on which basis the system defines the suitable technique for each contaminant under each concentration level. A total of six levels of technique selection for the different ranges and types of contaminants are implemented. The effluent concentration of each contaminant of current selection level is used as the influent concentration for the next level and this process is repeated till all the concentration of the waste contaminant meet its desired effluent quality. Thus the optimal treatment train is established.

The system requires to determine the various attributes of treatment techniques utilised and requires a specific rule-based system built from the treatability database. The rule tree is constructed for each type of contaminant so that the rules are formulated like, *if* the waste contaminant type is phenol *and* the influent concentration is between 100 mg/l to 1000 mg/l *then* the suitable technique is *AS* and the suitable technique is *SS*. Here *AS* stands for activated sludge and *SS* stands for *Steam Stripping* technique respectively.

After forming the suitable technique list, the preference technique is selected on the basis of defuzzification from the user defined membership functions for the construction cost and the performance efficiencies. The example of a verbally formulated rule can be *if the performance efficiency is "low" and the construction cost is "high" then the preference of the technique is "low"*. Nine rules are constructed for this part of the system. The expert system is applied to certain cases and has proved the ability to choose the optimal treatment train, however, the system requires to be enhanced further with additional knowledge extracted from the experts along with the use of more integrated tools to determine the cost effectiveness.

### 3.5.2 Inland water systems

#### Reservoir system operation

The fuzzy adaptive system (FAS) was used to determine the MISO control problem of reservoir operations (Bardossy and Duckstein, 1995). In the fuzzy modelling, the antecedents were the forecasted demand, incoming flow, present reservoir storage capacity and the time of the year. The consequence was a release from the reservoir. Only an observed hydrological data set is used for the training and verification of the fuzzy rules without any physically based model. The weighted counting algorithm was used for the generation of rule. The result was reasonable and it was found to be proving the robustness of the approach.

Panigrahi & Mujumdar (1997) applied the concept of a fuzzy rule-based system for determining the releases from a large-scale irrigation reservoir. The rule-base was built on the basis of the expert's knowledge. The MATLAB Fuzzy Logic Toolbox was used for the simulation. The storage of the reservoir, demand and inflow were the input variables to the fuzzy system. The triangular membership functions and the centroid method were used for the fuzzification and defuzzification respectively. The defuzzification was given by the following formula:

$$G = \frac{\sum_{i=1}^n y_i m_b(y_i)}{\sum_{i=1}^n m_b(y_i)} \quad (3.20)$$

where,  $n$  is the number of elements,  $G$  is a centroid of the truncated fuzzy output and  $m_B(y_i)$  is the membership value of the element  $y_i$  in the fuzzy output set  $B$ . The reliability and resiliency with respect to meeting the target releases were used as the performance criteria. The study concluded that the fuzzy logic could be a complement to the mathematical optimisation technique rather than a substitution.

The fuzzy logic approach was used for reservoir operations for flood control in the capital city of South Korea by Jae Hyun, S *et al*, (1999). The water release of reservoirs for the flood protection is based directly on the current water level and inflow without considering the predicted inflow. The fuzzy controller is developed to increase the efficiency in the flood control, while keeping the storage as maximum as possible for the supply of water. The results of the model with fuzzy controller on several big flood events are compared with the historical operation result. The fuzzy rule base is constructed with 7 categories depending on the water level and the storage capacity. Mamdani's min-max operation was used for the fuzzy implication and the centre of gravity method was used for the defuzzification. The results of the system using the fuzzy control shows successful reduction in water level and peak discharge by using preliminary releases. Also it can keep the maximum allowed release, while in the historical operation the allowed amount was violated many a time causing flood in the downstream area.

### Drought forecasting

Forecast of droughts in the U.S Great Plains by using the FRBM has been studied on the basis of atmospheric Circulation Pattern (CP) data of El Nino/Southern Oscillation (ENSO) phenomena (Pongracz *et al*, 1999). In this study, monthly value of Southern Oscillation Index (SOI) value is used, which is the most widely used indicator of warm and cold ENSO events. The drought indicating indices is Palmer Modified Drought Index (PMDI), which is based on the principle of a balance between moisture supply and demand without implying man-made changes. As input variables, the monthly relative frequency of daily CP types (for 6 different time lagged period) and the monthly values of SOI (for 4 different time lagged period) are defined and 5 triangular input membership functions for each variable are used for input fuzzification. Total of 18 overlapping output membership functions of PMDI are used for defuzzification of estimated output. The approach was used for 8 climate divisions of Nebraska and the result was found to be reasonable. The average correlation between the observed and estimated values is 75-80%. It confirms that the drought can be influenced by a large number of atmospheric, hydrological, agricultural phenomena, which were not considered in the study.

Also a possibility to apply the adaptive fuzzy controller for time-varying process (pilot plant) by direct adaptation without preliminary identification of the process is published by Venetsky *et al*, 1996. The method produces a control signal to compensate error due to a new reference signal or change in parameters in the system. It fuzzifies the error, performs inferencing and defuzzifies the result of inference to produce the signal. The measured error ( $err_y$ ) can be mapped into the vector of membership functions  $err_y \rightarrow (o_1, \dots, o_i, \dots, o_j \dots)$ . The fuzzification was done using the following equation:

$$o_i = e^{-\frac{(err_y - c_i)^2}{w_i}} \quad 1 < i < n; \quad (3.21)$$

where

$err_y$ - single state error

$c_i$  - position of  $i^{\text{th}}$  center of bell-shaped membership function

$w_i$  – width of  $i^{\text{th}}$  membership function

Then it can be said that the  $err_y$  belongs to fuzzy set ‘ $i$ ’ with membership  $o_i$ , or belongs to fuzzy set ‘ $j$ ’ with membership  $o_j$  ... etc, which indicates that more than one rule is fired. The centroid method is used for defuzzification. Compared to the conventional adaptive controller, the direct adaptive FLC performed better.

### Water quality analysis

As mentioned before, water quality problem has many inherent uncertainties and fuzzy logic approach is particularly suitable for uncertain problems with imprecise information. Water quality standards can be defined by smoother transitions from desirable to unsuitable quality levels. Comparative study on the use of fuzzy logic and multiple regression analysis for chemical water quality analysis and taste tests are carried out by Iwanaga *et al* (1997). The study was carried out on a data set with several varieties of high quality water in Japan.

The correlation analysis is carried out to investigate whether the standards for chemical evaluation confirm the national standard. The multiple regression analysis formula is created for four explanatory variables, overall hardness, organic material, iron ions and bicarbonate ions, which have a high correlation with the taste tests in all regions of the country. The multiple regression formula is used for judging the water quality for the other regions. And the fuzzy inference was applied to build a predictive model for judging water for all the regions in the country. The both results are compared and it was found that the performance of the fuzzy logic approach was better than the conventional predictive model.

For each of the four water quality factors, four triangular membership functions were assigned which can be described by three points (equation 3.22). The membership functions were tuned using neural net.

$$A_{ij}(x_j) = \begin{cases} \frac{x_j - a_{ij}}{b_{ij} - a_{ij}}; a_{ij} < x_j < b_{ij} \\ \frac{c_{ij} - x_j}{c_{ij} - b_{ij}}; b_{ij} < x_j < c_{ij} \\ 0; x_j \leq a_{ij}, x_j \geq c_{ij} \end{cases} \quad (3.22)$$

The initial values to tune the neural network are obtained from the value set by partitioning the definition domain based on the data distribution. For fuzzy inference rule the min operator is used (equation 3.23 and 3.24).

$$m_i = A_{i1}(x_1) \wedge A_{i2}(x_2) \wedge \dots \wedge A_{ij}(x_j) \quad (3.23)$$

$$y = \frac{\sum m_i \times T_i}{\sum m_i} \quad (3.24)$$

The estimation error  $D$  and decision coefficient is used as performance indices. Fuzzy logic approach performed better than the conventional regression analysis especially for water in other regions. The study has proven that the use of fuzzy logic approach can yield a more valid water quality judgement system than the multiple regression analysis. However, by gathering more data from many other regions the precision of the solution can be increased.

Chemical equilibrium process in a water system, called as hydro-geochemical system, is highly uncertain and has been studied by Schulz *et al* (1999). Fuzzy rule based model has been built to deal with the equilibrium of anaerobic sediment in small lake. The fuzzy model has shown improvement in analysing uncertainty in thermodynamic calculation schemes in chemistry.

#### Solute transport in unsaturated zone

FAS has been applied for solute transport simulation (Dou *et al*, 1999). The underlying physical processes for solute transport, which are described by differential equations, have been captured by fuzzy rule based system from the training set generated through the 2D model of water flow and solid transport in a saturated media. The laboratory soil column is used for experimentation. FAS optimised by least square algorithm was able to generalise the model result very well.

#### Stream water quality classification

Classifying water quality accurately is one of the major tasks for water quality management problem, particularly when there is an increasing concern about ecological impact of water pollution. Lee *et al* (1997) has investigated a classification of stream water quality, its toxicity and rarity using Fuzzy Rule-Based System (FRBS). Four classes of water quality are distinguished on the basis of ecological information. Physical and biological indicators are used to define the classes verbally. A total of 30 *if-then* rules are constructed by 4 distinct input membership functions for each indicator.

For the output, 7 membership functions for classification, 3 for toxicity and rarity were used. The application of the FRBS is compared with the application of normal expert system classification, which is based on the same rule-base. The result of comparison clearly shows that the use of FRBS can deal with the problem quite closer to the reality and can provide a smooth curved output instead of a stepwise graph obtained by the conventional expert system. The FRBS reduces an inaccuracy caused by crisp set boundaries.

#### Filling in missing data

FAS is used for filling gaps of incomplete precipitation data (Abebe *et al*, 2000) in a catchment of North Italy and compared with the ANN and statistical normal ratio method. The least MSE is obtained by FAS and the highest percentage within the tolerance target of 5% is obtained by MLP. In terms of robustness, normal ratio method was the best.

# Chapter 4 Neuro-fuzzy and hybrid approaches

## 4.1 Introduction

In the preceding chapter, some of the successful applications of the fuzzy logic approach were presented. The applications show the advantages of the fuzzy logic approach where the conventional model based approaches are difficult to be implemented. Unfortunately, with the increase in the complexity of the process being modelled, the difficulty in developing dependable fuzzy rules and membership functions increases. This has led to the development of another approach which is mostly known as neuro-fuzzy or fuzzy-neuro approach. It has the benefits of both neural networks and fuzzy logic and is attracting an army of researchers in this field. Defining the structure and size of neural networks and determining fuzzy rules and the membership functions systematically are main research areas concerning this AI technique (Lin, 1994). The neuro-fuzzy hybrid system combines the advantages of fuzzy logic system, which deal with explicit knowledge that can be explained and understood, and neural networks, which deal with implicit knowledge, which can be acquired by learning.

Basically, there are different hybrid development strategies that can be distinguished concerning the integration of the techniques. In total, there are five hybrid development strategies that have been identified (Medsker, 1995):

- stand alone (non-interactive, independent software components)
- transformational (same with the stand-alone, but the system begins with one of the technique and finishes with the other type of technique as a result of transformation)
- loose coupling (separate intelligent systems that communicate through data files, such as pre-processors, postprocessors, coprocessors and user interfaces)
- tight coupling (separate systems pass information through the memory resident data rather than the external files, basically the shared data structures that facilitate interactive problem solving via the independent agents)
- full integration (the systems share data structure and knowledge representation. Reasoning is accomplished either co-operatively or through a component designated as the controller)

Some of the structures of the neuro-fuzzy approach and their applications in the water related problems are briefly reviewed here.

## 4.2 Neuro-fuzzy hybrid system

Presently, the neuro-fuzzy approach is becoming one of the major areas of interest because it gets the benefits of neural networks as well as of fuzzy logic systems and it removes the individual disadvantages by combining them on the common features. Different architectures of neuro-fuzzy system have been investigated by number of researchers such as Lin (1994), Medsker (1995) and Jana (1996). These architectures have been applied in many applications especially in the process control.

Neural networks and Fuzzy logic have some common features such as distributed representation of knowledge, model-free estimation, ability to handle data with uncertainty and imprecision etc. Fuzzy logic has tolerance for imprecision of data, while neural networks

have tolerance for noisy data (Medsker, 1995). A neural network’s learning capability provides a good way to adjust expert’s knowledge and it automatically generates additional fuzzy rules and membership functions to meet certain specifications. This reduces the design time and cost. On the other hand, the fuzzy logic approach possibly enhances the generalization capability of a neural network by providing more reliable output when extrapolation is needed beyond the limits of the training data (Lin, 1994).

### 4.3 Neuro-fuzzy architecture

The neuro-fuzzy system consists of the components of a conventional fuzzy system except that computations at each stage is performed by a layer of hidden neurons and the neural network’s learning capacity is provided to enhance the system knowledge. Different architectures of neuro-fuzzy system are available out of which two prominent types are discussed here.

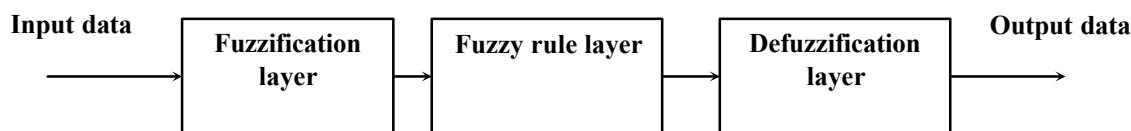


Figure 4.1: Schematic diagram of a neuro-fuzzy system

One possible architecture of a neuro-fuzzy hybrid system is shown in Figure 4.1 (<http://sunflower.singnet.com.sg/~midaz/Neufuzzy.htm>). The system contains the following three different layers:

- Fuzzification layer
- Fuzzy rule layer
- Defuzzification layer

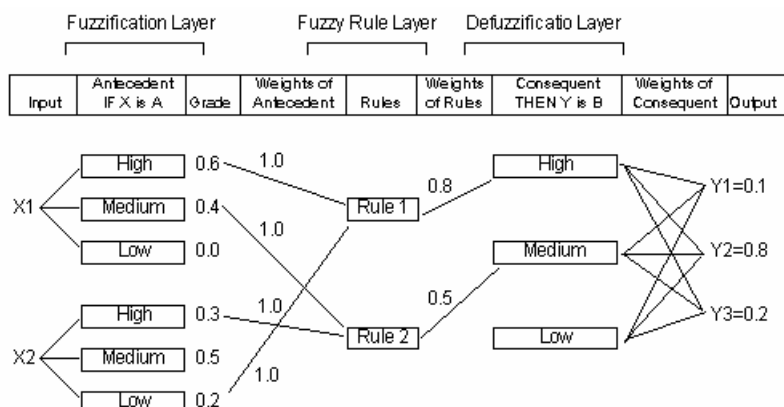


Figure 4.2 Neuro-fuzzy system structure (<http://sunflower.singnet.com.sg/~midaz/Neufuzzy.htm>)

In a *fuzzification layer* each neuron represents an input membership function of the antecedent of a fuzzy rule (Figure 4.2). In a *fuzzy inference layer* fuzzy rules are fired and the value at the end of each rule represents the initial weight of the rule, and will be adjusted to its appropriate level at the end of training. In the *defuzzification layer* each neuron represents a consequent proposition and its membership function can be implemented by combining one or two sigmoid functions and linear functions. The weight of each output link here represents



the centre of gravity of each output membership function of the consequent and is trainable. After getting the corresponding output the adjustment is made in the connection weights and the membership functions in order to compensate the error and produce a new control signal.

The neuro-fuzzy control architecture proposed by Lin (1994) has five layers, each of which performs fuzzy rule-based system operations (Figure 4.3). This belongs to the fully integrated hybrid intelligent system. Computer simulations satisfactorily verified the performance of this structure of a neuro-fuzzy control. Like a typical neural network, each node of this neuro-fuzzy connectionist model has connecting weights for incoming variables and the weighted sum is transformed through the activation function as enumerated in equation 2.2. Let us consider a single node from each layer as shown in Figure 2.4. Let us denote  $f$  as the transfer function (which is a weighted sum in a normal neural network),  $a$  as the activation function,  $u_p^k$  as the inputs to the node,  $w_p^k$  as the input weights and  $o_i^k$  as the outputs of the node, where  $k=1 \dots 5$ ,  $p$  and  $i$  are the total number of the input and output variables respectively. The process of each node can be described as follows:

- Layer 1: consists of the input nodes and directly transmits the input linguistic variables to the next layer. The link weight,  $w_i^1$ , at the layer one is unity.

$$f = u_i^1 \quad a = f \quad (4.1)$$

- Layer 2: is the input term layer where each node acts as a membership function and represents the terms of the respective linguistic variables. Instead of calculating a weighted sum as in a normal neural network model, it calculates, for example, in terms of a bell-shaped fuzzy membership function,

$$f = M_{x_i}^j(m_{ij}, s_{ij}) = -\frac{(u_i^2 - m_{ij})}{s_{ij}^2} \quad a = e^f \quad (4.2)$$

where,  $m_{ij}$  and  $s_{ij}$  are the centres (or mean) and the width (or variance) of the bell-shaped function of the  $j$ th term of the  $i$ th input linguistic variable  $x_i$ . The link's weight at the second layer ( $w_{ij}^2$ ) can be interpreted as  $m_{ij}$ . Also a set of nodes can represent a membership function in which case the nodes act as a standard neuron model as in equation 2.2.

- Layer 3: this layer is a fuzzy rule layer where each node represents one fuzzy rule. The links in this layer are used to perform antecedent matching fuzzy logic rules and also the fuzzy *AND* operation. The link weight,  $w_i^3$ , at the third layer is also equal to unity.

$$f = \min(u_1^3, u_2^3, \dots, u_p^3) \quad a = f \quad (4.3)$$

- Layer 4: the links to this layer define consequences of the rule nodes. The links in the third and fourth layer, in association, function as a fuzzy inference engine. The nodes in this layer have two operation modes: the *down-up* transmissions and the *up-down* transmission mode. In the *down-up* mode, the links in the layer should perform the fuzzy *OR* operation to integrate the fired rules which have the same consequence.

$$f = \sum_{i=1}^p u_i^4 \quad a = \min(1, f) \quad (4.4)$$

The link weight,  $w^4_i$ , at the fourth layer is also equal to unity. In the *up-down* mode, the nodes in the layer 4 and the layer 5 function exactly the same way of the nodes in the layer 2 except that only a single node is used to perform a membership function for the output linguistic variables.

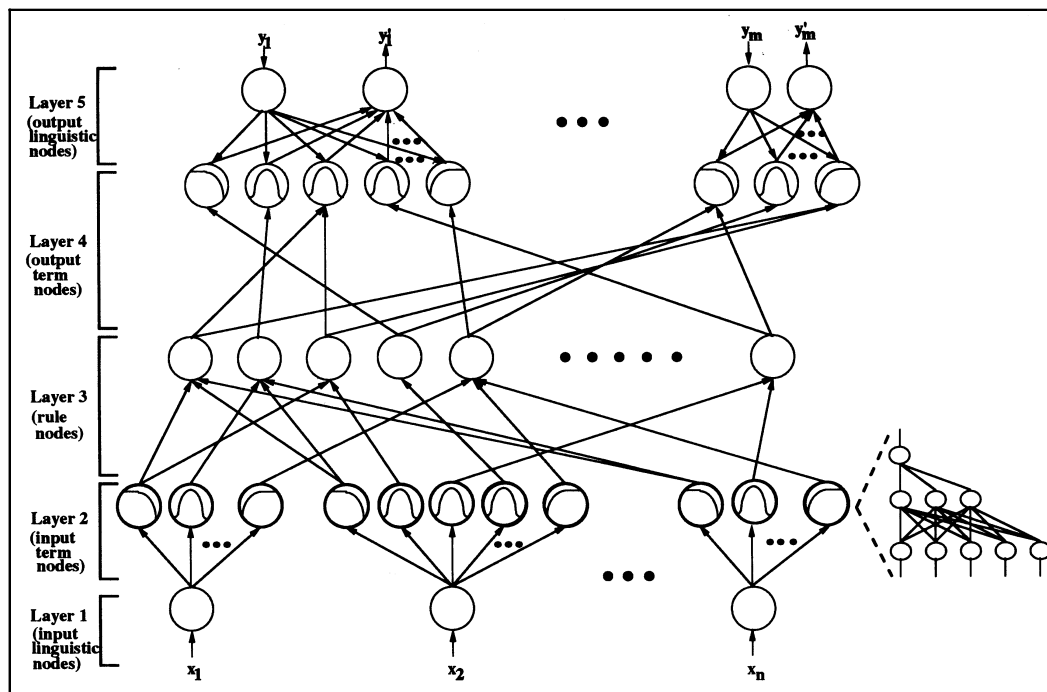


Figure 4.3: Neuro-fuzzy connectionist model (Lin, 1994)

- Layer 5: is an output linguistic layer, which has two nodes for each input variable. The first type of node is used to feed the networks for training it again.

$$f = u_i^5 \qquad a = f \tag{4.5}$$

The second type of node is used for deriving a real output or a decision signal.

$$f = \sum w_{ij}^5 u_i^5 = \sum (m_{ij} s_{ij}) u_i^5 \qquad a = \frac{f}{\sum s_{ij} u_i^5} \tag{4.6}$$

The details of the above learning algorithm for this structure can be found in Lin (1994).

#### 4.4 Other hybrid approaches

There are some other approaches of hybrid intelligent systems, which are combinations of different AI (machine learning) techniques such as expert systems and neural networks, fuzzy logic and expert systems, genetic algorithms and neural networks, genetic algorithms and fuzzy systems genetic algorithms and expert systems etc. It is not intended to cover all the hybrid approaches in detail. Further reading of such approaches can be found in Medsker (1995), Goonatikale and Khebbal (1995).

Many of the possible combinations of different AI approaches have been explored. According to the publications between 1988 and 1994, the most studied hybrid approach is expert systems and neural networks, expert systems and fuzzy logic (Medsker, 1995). The neuro-fuzzy approach is the third in the list. The latest publications show that the applications of neuro-fuzzy system and the neural networks with genetic algorithms are increasing, while application of expert systems with neural networks and the expert systems with fuzzy logic is decreasing.

## 4.5 Application of neuro-fuzzy systems

### 4.5.1 Drinking water systems

#### Estimation of state of water distribution network

In a water distribution network, the system state is estimated on the basis of the telemetry measurements and the prediction of consumption (called as pseudo measurements). Both of the requirements can have high level of uncertainties. The above measurements are used to determine the state of the system through a mathematical modelling, of which the performance is not always guaranteed due to uncertainties in the input parameters of the model. There are two types of faults or uncertainties in such a system: measurement errors caused by equipments and topological errors caused by faults due to the leakage and wrong valve status. Measurement errors are not correlated and thus the measurements with error can be discarded. But the detection and identification of topological errors are still not studied comprehensively. The Generalised Fuzzy Min-Max Neural Networks (GFMM NN) approach for clustering and classification is applied for this purpose (Gabrys and Bargiela, 1999). This is a fully integrated hybrid structure.

The neuro-fuzzy recognition system is used to identify and detect a leakage in the system. The training data set is generated by the system state estimation procedure combined with the Confidence Limit Algorithm (CLA) for the quantification of inaccuracies of system state estimation due to uncertainties in input data. The state estimation procedure is based on the mass balances in each node and the specific measurements taken on the node. The neuro-fuzzy recognition considered here is based on the hyperbox fuzzy sets. The hyperbox defines a region of  $n$ -dimensional pattern space and is defined by its min-max points. The hyperbox created during the training can represent a distinctive state of the system such as the normal operating state, a leakage between two nodes etc.

A neural network that implements the GFMM clustering/classification algorithm is a three-layered feedforward network. The input layer has  $2 \times n$  number of nodes, two for each of the  $n$  dimensions of the input pattern. Each node in the second layer represents a hyperbox fuzzy set. The connections in the 1<sup>st</sup> and 2<sup>nd</sup> layer are min-max points and the transfer function is the hyperbox membership function. Each node in the third layer represents a class. The connections between 2<sup>nd</sup> and 3<sup>rd</sup> layer are binary values: 1 if the second layer hyperbox fuzzy set is a part of the class represented by the output layer node and 0 otherwise. They are stored in a matrix form. The output can be either fuzzy or crisp.

Generation of the training data set of the networks is done in 3 stages: simulation of the state, estimation of accurate measurements and the CLA. Leakage of the system is simulated as a demand between two nodes and not as a pressure difference. Reservoir inflows and the other

network consumption are adjusted to compensate the additional demand. The wrong operation of a valve is simulated in a way that the valves those are usually open, remain as closed.

The recognition system developed is a two-level system where the first level is to distinguish the typical behaviour of the system (such as night load, peak load etc.) and the second level is to detect the anomalies. The result of this approach shows that the neuro-fuzzy system can be trained successfully for the estimated system-state as well as the residuals with their confidence limit. Both have advantages and disadvantages, however, the simulation of a system based on the estimated system state gives better results in terms of accuracy.

### Tank level control

The stand alone structure of the hybrid system is applied for controlling the tank level in solvent dewaxing (oil refinery) plant (Tani *et al*, 1994). The controlling purpose is to keep the tank level stable and to change the outflow rate from the tank as smoothly as possible in order to keep the whole process normal and continuous. The whole dewaxing process itself is difficult to be controlled because of uncertainties and complexities, therefore, usually is controlled by the experts. The difficulty occurs due to the following reasons: the inflow rate to the tank varies with oil filter plugging, feed oil is switched on frequently, the heater has a limit in the change of the flow rate, two different states to control (steady and transient) etc. The steady state is when the tank level goes down and up periodically by stopping and washing one of the oil filters. A transient state occurs when in addition to the above, the feed oil is changed completely, which results the tank level to drop down rapidly.

In order to deal with these states, the neuro-fuzzy controller is built with the following three components:

1. a statistical component to calculate long time tendencies of the flow rate from the historical operational data.
2. a correction component (fuzzy logic) for compensating the flow rate from statistical component to stabilise the tank level. The rule base is built on the basis of the experts' knowledge.
3. a prediction component (neural networks) to predict the inflow rate when the oil is being changed. That is the target of the fuzzy logic controller.

Application of the neuro-fuzzy controller smoothes the tank levels not only in a steady state, but also in transient state when the feed oil is changing. For example, applying the neuro-fuzzy system the tank level ranges between 35%-75%, while on the basis of experts' knowledge it ranges between 30%-80%.

### 4.5.2 Sewerage systems

#### Waste water treatment process

Detection of faults in anaerobic process using the fuzzy-neural network has been considered by Steyr *et al*, (1997). The process is very complex as well as unstable and depends on the incoming flow rate, influent organic load etc. There are problems that are less predictable such as pipe clogging which causes an increase in valve opening, foam forming which changes gas flow rate etc. Besides, there are local controllers used to control the individual

processes. But there is no technique using on-line measurements and handling an ill-defined process as a whole.

The structure of the neuro-fuzzy system is as follows: The measured signals are transferred into fuzzy variables depending on whether the variable is deviated from the mean value or not. By using additional fuzzy rules, the occurrence of a faulty situation in the system is determined.

Another loosely coupled hybrid model of an ANN and fuzzy logic has been applied for diagnosis of anaerobic treatment plant (Steyer *et al*, 1997). The raw data has been processed by fuzzy logic to build a pattern vector (training data), where training data set is classified into pre-specified categories indicating the state of the system. An ANN is then used to classify the process states and to identify the faulty and dangerous states. The hybrid model recognises the situations caused by pipe clogging, foam forming and bad temperature regulation. The approach can be seen as a tool able to handle with large number of problems in a simple frame.

#### 4.4.3 Inland water systems

##### Reservoir system operation

The neuro-fuzzy system has been found to be a suitable approach for a multi-purpose reservoir system operation (Hasebe *et al*, 1999). The study concentrates on the application of a fuzzy neural network (stand-alone hybrid structure) and a fuzzy system for reservoir operation and presents a comparison of results obtained. The mathematical expression of dam operations is difficult and somewhat vague because of the presence of many different constraints which need to be considered. On the other hand the inflow can be predicted on the basis of abundantly available hydrological information within the catchment. The composed system is applied for determining the operation of a reservoir for irrigation and flood control purposes.

The operation line is determined on the basis of the water level in the reservoir, changing inflow, inflow and precipitation coupled with actual historical operation. The neural network used 7 input variables (rainfall, river discharge, predicted flow, changing inflow, water level and release discharge), one hidden layer with three nodes (response to dam basin, discharge and the state of the reservoir) and an output layer with one neuron (describing release of discharge, storage volume or conservation of water level in the reservoir).

For irrigation purposes, the fuzzy control and neuro-fuzzy control give smoother release of discharge. The fuzzy control gives better result in terms of storage volume. For flood control purpose during the typhoon both the controllers give the maximum release, however, the fuzzy neural networks give higher peak value than the fuzzy controller.

##### Beam balancing system

The above principle was investigated with a six-layered neural network, in which each layer performs specific actions to represent the fuzzy inference mechanism (Jana, 1996). The six layers are the input layer, a fuzzification layer, two layers for fuzzy inferences, a defuzzification layer and an output layer. The designed controller is applied for experimental fluid beam balancing system, which balances an unstable beam contained in two tanks, one at

each end pumping back or forward from the tanks. The problem was formulated as Multiple Input and Single Output (MISO) problem and the real-time control was evaluated against a PID controller. After a short simulation, the algorithm gave reasonable results compared to the PID controller and further investigation of RTC was suggested.

## Chapter 5 Discussion, conclusions and recommendations

### 5.1 Discussion

In the present study, the applicability of some *data-driven* techniques stemming from *Artificial Intelligence*, *Data Mining* and *Soft Computing* in the integrated water management field is explored. The data-driven techniques considered are *artificial neural networks (ANN)*, the fuzzy logic approach (mainly, *fuzzy rule-based systems, FRBS*), and the *neuro-fuzzy* approach. Here we summarise the observations.

#### Traditional vs data-driven models

Traditional physically-based models use mathematical expressions to describe a physical process. The desired solution is also achieved mathematically. Let us take the example of a regional water system containing polder(s) and pumping station(s). A physically-based model uses a large number of system state variables to describe the physical process. Some representative system state variables can be water level, precipitation, predicted precipitation, evaporation, soil moisture condition, groundwater table etc. The water system is expressed through mathematical equations such as continuity equations to describe the physical process. A mathematical optimisation problem is solved to determine the pumping strategy. A complete knowledge of the physical process is essential.

On the contrary, the AI (machine learning) related techniques require input-output data representing any process being modelled. For a water system, the input parameters can be water level, precipitation etc. The output parameter can be the pumping strategy. Time series data of the input and output parameters are arranged to train a data-driven model. When we have sufficient data of the process, a reliable model may be developed. Once the model is developed it can be used to determine outputs based on new inputs.

It is noteworthy that to develop the data-driven model of the water system described above, a thorough knowledge of the process is not required. This gives an edge over the physically-based models for complex processes which are difficult to be expressed through mathematical equations with a reasonable degree of accuracy.

At some occasions, we may have a dependable physically-based model which requires high computing time to solve the problem. The AI (machine learning) techniques are used to approximate the conventional model for saving computing power and bring in faster computing speed. At most of the occasions, data-driven models run at a much faster speed than the conventional models.

In general, the advantage or benefits of using ANN and fuzzy logic concepts for water related problems could be described as *the possibility to complement the traditional physically-based methods, or to replace them if they fail*.

### Source of data

Measured data is the main source of data for developing a data-driven model. If developing a data-driven model of a water system consisting of polders and pumping stations is considered then time series data of the selected input and output variables are required. The data should contain all possible variations of the input and the output parameters. If precipitation is one of the input parameters, then a hydrological database containing all possible variations in the hydrological conditions is required for developing a dependable data-driven model. This is true for all other input and output parameters.

Another possible source of data can be a physically-based model. The model is run to generate data of possible conditions. Subsequently, this data is used to develop an AI approximation of the physically-based model. Of course, the AI techniques can only pick up the required behaviour if the available physically-based model is accurate and describes the relevant process in detail.

### Artificial neural network

Neural networks are said to perform a human-like reasoning, learn the attitude and store the relationship of the processes on the basis of a representative data set that already exists. ANN consists of distinct layers of processing units and their weights connecting each unit (node). Depending on the structure of the network, usually a series of connecting neuron *weights* are adjusted (called training) to fit a series of inputs to another series of known outputs. Once the training is performed, the verification is very fast. An ANN is distinguished by the learning principle, the way the information flows within the network, the structure of the layers etc.

Based on the instant review, the choice of a technology to a specific field in the water management scenario is summarised in Table 5.1. Among the different ANN structures, the *Error Backpropagation* (EBP) network is widely used and bears a track record of reasonable success. *Self-organizing Feature Map* (SOFM) networks are often used for the classification and identification purpose. Some comparisons show that the *Brain-State-in-a-Box* (BSB) network may be preferred to the SOFM network (Vermeersch *et al.*, 1999). The *Radial Basis Function* (RBF) networks are applied to a number of water management problems. Although the RBF network learns faster than other networks, its accuracy of solution is often not satisfactory.

The applications of the *Recurrent Neural Network* show its ability to work with noisy data set. However, the recurrent neural network is not applied as extensively as the EBP network. The *Principal Component Neural Network* (PCNN) itself is not applied for water management field, however, number of published applications on classification and image analysis in the water management field using *Principal Component Analysis* (PCA) show the possible potential of this type of neural networks. The PCNN is successfully applied in other fields such as gender recognition (Diamantaras and Kung, 1996).

### Fuzzy logic approach

The fuzzy logic approach has a long history of industrial applications compared to ANN, particularly in the field of process control. A fuzzy logic approach is particularly a preferable tool for dealing with problems with uncertainties and imprecise information. The main phases



in a fuzzy logic system are *fuzzification*, *fuzzy inference* and *defuzzification*. It works on the basis of the *fuzzy rules* consisting of arguments coupled with logical operators (*AND*, *OR*,

Table 5.1: Application preference of considered techniques

	Techniques	Preference for application	
		suitable	not suitable
1	<b>Neural networks</b> - Error backpropagation  - Radial basis function  - Recurrent  - SOFM  - Principal component NN	Model generalization, function approximation, pattern recognition, prediction, identification, classification, optimal control (approximate optimal solution) etc  Generalization, approximation, prediction, identification pattern recognition, prediction, identification, classification, optimal control, etc  Complex dynamic processes with noisy and less number of training data, identification, classification, pattern recognition  Identification, classification, pattern recognition  Identification, classification, data reduction (dimension) or feature extraction, noise reduction in data	Problems where uncertainty involved  Problems with less number of training data, uncertain problems  Problems where uncertainty involved  Model generalization, prediction, approximation, generalization  Prediction, model generalization, approximation
2	<b>Fuzzy logic approach</b> - Fuzzy inference  - Fuzzy adaptive system	Processes with uncertainty, optimal control (combined with the expert knowledge), maintenance, identification, classification  Generalisation, approximation, identification, classification, dynamic system control	Prediction
3	<b>Neuro-fuzzy hybrid systems</b>	Complex system control, maintenance, identification, classification, system analysis, diagnosis	

*NOT* and *XOR*). The rules are verbally formulated such as *IF* the condition is fulfilled *THEN* the consequence is *true*. The rules can be elicited from the experts or generated on the basis of the available training data. A set of given data is fuzzified according to the membership functions assigned to it. The truth value of the fuzzified data corresponding to the fulfilment of conditions is determined and is called the *degree of fulfilment* (DOF). Then the fuzzy rules are responded in different combinations to get a fuzzy output from the fuzzy inference engine. The output is defuzzified in order to get a crisp output at the end.

The *Fuzzy Rule Based System* (FRBS) is distinguished into two parts in this review: *Fuzzy Inference*, which works on the constructed rule-base, and *Fuzzy Adaptive System* (FAS), which builds a rule-base on the basis of a given training data set. The fuzzy inferences are

applied successfully for identifying the optimal control actions in wastewater treatment plants, in determining the optimal dose of chemicals in wastewater treatment plants and in determining leakage in a water distribution network. They are often used in combination with the expert's knowledge. FAS have been applied for drought prediction, in determining the optimal control action of polder pumping stations and filling in the gap of measured data. It has proven its ability to learn as good as the neural networks.

### Neuro fuzzy approach

The neuro-fuzzy approach is comparatively a new field and is a growing area of research activities. A number of successful applications of this technique in the water management field have been published. The neuro-fuzzy systems combine the advantages of fuzzy logic system, which deals with explicit knowledge that can be explained and understood, and neural networks, which deal with implicit knowledge, which can be acquired by learning. The neuro-fuzzy system has been applied successfully for detecting and identifying faults due to any measurement error, leakage or wrong valve status in a water distribution system.

Based on the instant review, the suitability of an AI technique to the problems of water management field is summarised in Table 5.1.

### Applications of AI techniques to water management field

Like in many other fields, these AI techniques are also applied widely in the water management problems and are becoming the mainstream research subjects nowadays. In the present report, the applications of the Artificial Intelligence (AI) techniques namely ANN, fuzzy logic approaches and neuro-fuzzy approaches in an integrated water management field have been reviewed. In total 85 papers, 14 theses and 15 books were reviewed. An exhaustive list of the applications and their efficacy is presented in Table A.1 (see Appendix). As can be seen from the summary Table A.1, the applications of these techniques have shown a reasonable accuracy for most of the cases. From the review, it is possible to conclude that any logical relationship can be generalised and approximated by neural networks with reasonable accuracy if a set of input-output data is available. On the other hand, the FRBS is able to deal with the processes which are non-linear and bears a degree of uncertainty. The applicability of these techniques is based on a number of parameters but they mostly depend on the quality and quantity of the available data set.

The important sub-disciplines of the integrated water management field where applications of the AI techniques are noticed are highlighted in the following:

#### **Assessment**

##### **a. Resources or quantity assessment**

- rainfall runoff modelling
- rainfall prediction
- urban runoff prediction
- drought forecasting
- groundwater flow simulation
- river bed evolution
- determining the erosion of groyne field sediment
- simulation of hydro morphological processes
- Q-h relationship (rating curve)

- Reservoir inflow prediction
- Classifying discharge pattern
- Classification of storm and wind events

#### **b. Water quality management**

- sewer water quality
- quality of urban storm water
- groundwater quality survey
- water quality evolution in a pipe network
- water quality evolution in natural flows
- water quality classification of river catchment
- identification of pollutant source in surface water
- identification of groundwater contaminant

#### **c. Ecological relations**

- species mass prediction
- determining the eco-regions
- classification of ecological data

#### **Planning`**

- community water demand prediction
- reservoir operation planning
- tidal water current prediction
- CSO simulation

#### **Operation and real-time control**

- optimal control of water resources system
- optimal reservoir system operation
- CSO minimization
- water treatment process control
- determining dosage of treatment
- selection of treatment plant procedure
- determining tidal water level
- water bath temperature control of reactor

#### **Maintenance**

- fault detection or distinction in distribution network
- determining leakage point in the networks

Reviewed applications of these techniques reveal that they are applied mainly to operation, assessment and operational planning purposes in the water management field (Figure 5.1). To a lesser extent the techniques have been applied to maintenance purposes. No publication on the use of these techniques in extremely complex processes like dredging activities is published yet. More detailed presentation of the percentage wise applications of the AI techniques in terms of the total reviewed papers can be seen in Table 5.2.

ANN is extensively applied for assessment purposes like rainfall-runoff modelling, water quality prediction in natural flows, approximating ecological relations. It has also been applied for the optimal reservoir operation. A remarkable number of publications on the application of the fuzzy logic approach for process control in wastewater treatment plants to

determine the optimal control actions are available. Problem of real-time optimal operation for water related systems has been investigated by using neural networks, fuzzy logic approach and neuro-fuzzy approach. The neuro-fuzzy systems have been applied only for operation and maintenance problems.

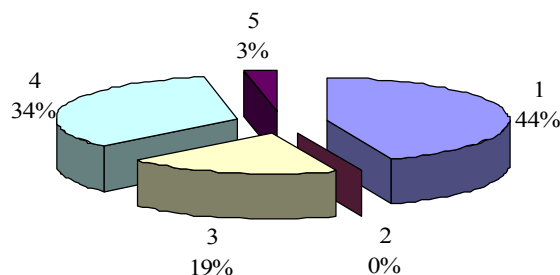


Figure 5.1 Percentage of ANN and fuzzy logic for water management  
 1- assessment, 2- design, 3- planning, 4- operation, 5- maintenance

Table 5.2: Percentage of considered techniques applied in water management

Activity	ANN (%)	Fuzzy logic (%)	Neuro-fuzzy (%)
1 Assessment	32.4	11.8	0.0
2 Design	0.0	0.0	0.0
3 Planning	14.7	4.4	0.0
4 Operation	13.2	16.2	4.4
5 Maintenance	0.0	0.0	2.9
Σ	60.3%	32.4%	7.3%

## 5.2 Conclusions

- ANN and Fuzzy logic techniques are applied to a large number of problems in the water management field. They are mostly applied in the following areas:
  - Quantity assessment such as rainfall runoff modelling, urban runoff prediction, rainfall prediction etc.
  - Water quality management such as sewer water quality, urban storm water quality, water quality in a pipe network etc.
  - Ecological relations such as mass prediction of species etc.
  - Planning such as community water demand prediction etc.
  - Operation and real time control such as determining pumping strategy in polder water level maintenance, optimal reservoir operation etc.
  - Maintenance such as fault detection a water distribution network etc.
- The techniques have been most extensively applied for the assessment and operation purposes, and to a lesser extent to planning, maintenance and designing problems of water resources management. In general, the number of application of the data-driven techniques in the water sector is not so high as in other fields.
- One important advantage of the considered techniques over the physically-based models is that the domain specific knowledge of the process being modelled is required to a

lesser extent. In the case of the physically-based models, the process has to be expressed through mathematical expressions. An in-depth knowledge of the process is required to do that. At some occasions, the process may be too complex to be described mathematically or the physical process may not be well known. In that case, the data-driven models based on the measured data can solve the problem.

4. The data-driven models act as a complement to the physically-based models. It may also be used to replace the physically-based models. At other occasions, where the physically-based models cannot be developed due to a lack of proper knowledge of the process, the AI techniques provide the way to develop a model of the process.
5. It may take considerable time to develop a data-driven model based on these AI techniques. But once the models are developed, they usually run much faster than the physically-based models.
6. In developing a data-driven model, preprocessing of data and selecting the right input and output variables is an important task. For example, for a water system consisting of polder(s) and pumping station(s) then detailed statistical analyses are required to determine the right input variables for developing a data-driven model of the system. The right input variables may be water level, precipitation etc. selected from the set of system state variables. The output variable may be the pumping rate.
7. The efficacy of a data-driven model depends upon the range and quality of the source data. If the source data does not cover all possible state conditions, then the data model may not perform well.
8. The techniques described in this report (and many other machine learning techniques) are in fact mathematical and modelling apparatus that has a general nature and can be applied practically in any area (as, for example differential equations). There are many ways of applying machine learning and data mining techniques, and selecting the right AI technique to a specific problem is important. Applicability of a certain technique to a specific problem has to be explored first before attempting to develop a data-driven model. This is often more "art of modelling" than the science of it.
9. The field of AI techniques is an active research area with the new developments coming up to remove previous bottlenecks. Data acquisition is becoming more and more convenient day by day and the computing speed is much less of a problem than before. These two phenomena open up new horizons to using AI-related techniques (machine learning, data mining, soft computing) in various fields especially in the area of water resources management.

### 5.3 Recommendations

1. The applications of the data-driven techniques reviewed in this report show the immense potential of these techniques in developing models of any physical process such as that within a water system. With the day-by-day increase in the computing speed of personal computers and an increase in the convenience of data collection, these techniques are going to be the key technologies in modelling practices. The applications of these techniques in the water sector are still less compared to the applications in the other

sectors. In general, more research on exploring new areas of application of the AI techniques in the water sector is recommended.

2. The AI techniques described in this report is not an exhaustive list. There are many other suitable techniques such as: *support vector machine, chaos theory, reinforcement learning, nearest neighbour method, decision rules* etc. Every method has its advantages and disadvantages and is well suited to some specific problems. A recent study at IHE shows that reinforcement learning can be successfully applied to determine the optimal pumping strategy in a regional water system. The applicability of these techniques to the operational water management may be explored.
3. The present study proves the potential of using these techniques in the water sector and accordingly it is felt right to go ahead with exploring these techniques in the water system control. A further detailed research on the application of ANN, the fuzzy logic approach, the neuro-fuzzy approach, reinforcement learning and the related techniques to the operational water management is recommended.

## References

1. Aafjes, J, Verberne, A.J.P, Hendrix, L.J.W, Vingerhoeds, R.A, 1997, *Voorspellen van het drinkwaterverbruik met een neuraal netwerk*, H<sub>2</sub>O (30), nr. 15, pp. 484-487 (in Dutch)
2. Abarhanel, H.D.I, 1996, *Analysis of observed chaotic data*, Springer\_Verlag, NewYork, Inc
3. Abebe, A.J, Solomatine, D.P, Venneker, R.G.W, 1999, *Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events*, paper sent for publication of Journal of Hydrological Sciences.
4. Abebe, A.J, Price, R.K, 2000, *Application of neural networks to complement physically based hydrodynamic models*, Proc. 4<sup>th</sup> International Conference on Hydroinformatics, Iowa City, USA, Balkema, Rotterdam
5. Amdisen, L.K *et al*, 1994, *Model-based control - a hydroinformatics approach to real-time control of urban drainage system*, Journal of Hydraulic Research, vol. 32
6. Aoi, T, Okinawa, Y, Hagiwara, K, Motomura, K, Iwaihara, E, Imai, M and Serizawa, Y, 1992, *A direct ammonium control system using fuzzy inference in a high load biological denitrification process treating human excreta*, Wat, Sci. Tech, vol. 26, no. 5-6, pp. 1325-1334
7. Baffaut, C and Chameau, J.L, 1990, *Estimation of pollutant loads with fuzzy sets*, Civil Engineering Systems, vol. 7, no. 1, pp 51-61
8. Bardossy, A and Duckstein, L, 1995, *Fuzzy rule-based modelling with applications in Geophysical, biological and engineering systems*, CRC press Inc
9. Bazartseren, B, 1999, *Use of Artificial Neural Networks and Fuzzy Adaptive systems in controlling polders water level in the Netherlands*, MSc thesis HH352, IHE, Delft
10. Beale, R and Jackson, T, 1990, *Neural computing: an introduction*, J W Arrowsmith Ltd, Bristol
11. Bhattacharya, B, 1998, *The use of Neural Networks in Optimal control of water resources systems*, Individual study report, IHE, Delft
12. Bhattacharya, B, 2000, *Application of artificial neural network in stage-discharge relationship*, Proc. 4<sup>th</sup> International Conference on Hydroinformatics, Iowa City, USA, Balkema, Rotterdam
13. Bourke, M.M & Fisher, D.G, 1996, *A predictive Fuzzy relational controller*, Proceedings of 5<sup>th</sup> IEEE conference on Fuzzy systems, vol.2, pp 1264-1270
14. Breitscheidel, A, Cser, J and v.d Veer, P, 1998, *Applicability of neural networks in hybrid water models*, proceedings of Hydroinformatics '96 conference, Babovic, V & Larsen, L.C eds., Balkema, pp 749-752
15. Crigg, N.S, 1996, *Water resources management: Principles, regulations and cases*, McGraw-Hill, New York
16. Diamantaras, K.I and Kung, S.Y, 1996, *Principal Component Neural Networks: Theory and applications*, John Wiley & Sons. Inc, New York
17. Dibike, Y.B, 1997, *Use of artificial neural networks in model approximation*. MSc thesis, IHE, Delft, The Netherlands
18. Dou, C, Woldt, W and Bogardi, I, 1999, *Fuzzy rule based approach to describe solute transport in the unsaturated zone*, Journal of Hydrology, vol. 220, pp 74-85
19. Gautam, D.K, 1998, *Regressive neural networks for modelling time series*, MSc thesis HH333, IHE, Delft
20. Gautam, D.K, 1999, *Classification of wind events Using Kohonen networks*, Joint workshop on Neural networks in Civil Engineering, Delft
21. Golden, R. M, 1993, *Stability and optimization analysis of the generalized Brain-State-in-a-Box neural network model*, Journal of Mathematical psychology, vol. 37, no.2, pp. 282-298
22. Gong, N, Ding, X, Denoeux, T, Bertrand-Krajewski and Clement, M, 1996, *STORMNET: A connectionist model for dynamic management of wastewater treatment plants during storm events*, proceedings of special IAWQ Conference on Sensor in Sewers, Copenhagen, Denmark, pp. 247-256
23. Goonatikale, S and Khebbal, S, 1995, *Intelligent hybrid systems*, John Wiley & Sons, Great Britain

24. Götz, R, Steiner, B, Sievers, S, Friesel, P, Roch, K, Schwörer, R and Haag, F, 1998, *Dioxin, dioxin like PCBs and organotin compounds in the river Elbe and the Hamburg harbour: Identification of sources*, Water Science and Technology, vol. 37, no. 6-7, pp 207-215
25. Gurney, K, 1999, *Neural nets*, <http://www.shef.ac.uk/psychology/gurney/notes/contents.html>
26. Gybris, B and Bargiela, A, 1999, *Neural networks based decision support in presence of uncertainties*, Journal of water resources planning and management, Sept/Oct, pp. 272-280
27. Hanisch, W.S and Pires, E.C, *A neural network model to predict operational parameters of a wastewater treatment plant*, proceedings of Hydroinformatics '98 conference, Babovic, V & Larsen, L.C eds., Balkema, Rotterdam, pp 765-770
28. Harris, C.J, 1994, *Advances in Intelligent control*, Taylor & Francis Ltd, London
29. Hasebe, M, Kumekawa, T and Kato, T, 1999, *Comparison of fuzzy system and fuzzy-neural network systems in respect of control system for reservoir operation*, proceedings of International conference in Controlling the complex system, pp 6-11
30. Hassan, K.M, 1995, *Real-time control of Hydraulic structures: Predictive control using Artificial neural networks*, MSc thesis, IHE, Delft
31. Hillier, F.S and Lieberman, G.J, 1995, *Introduction to mathematical programming*, McGraw-Hill, Inc
32. Hirota, K (ed), *Industrial applications of Fuzzy technology*, Springer-Verlag, Tokyo, 1993
33. Hou, S.L and Ricker, N.L, 1992, *Minimization of combined sewer overflows with Fuzzy Logic Control*, IEEE Proceedings of 1<sup>st</sup> International Conference on Fuzzy Systems, San Diego, CA, pp 253-260
34. <http://sunflower.singnet.com.sg/~midaz/Neufuzzy.htm>, *NeuroFuzzy computing*, New Wave Intelligent business Systems, NIBS, Inc, NIBS Pte Ltd Technical report TR-960308
35. Iwanaga, C, Sasaki, K, Hamoaka, T, 1997, *Evaluation of the Quality of Water Using Fuzzy Reasoning and Multiple Regression Analysis*, Japanese journal of Fuzzy theory and systems, vol. 9, no.3, pp. 291-306
36. Jae Hyun, S, Jitae, K, Jun-Haeng, H and Woncheol, C, 1999, *A study on the reservoir operation for flood control using fuzzy control technique*, Proc. XXVIII IAHR congress, Graz, Austria
37. Jana, A *et al*, 1996, *Real-time Neuro-Fuzzy control of a nonlinear dynamic system*, Proceedings of the 1994 Biennial Conference of the North American Fuzzy Information Processing Society, pp 210-214
38. Julien, B, 1994, *Water quality management with imprecise information*, European journal of Operational Research, vol. 76, pp. 15-27
39. Kalker, T, Babuska, R, Roeleveld, P, Ruland, M and van Goor, C, 1999, *Toepasbaarheid van fuzzy control bij het zuiveren van stedelijk afvalwater*, H<sub>2</sub>O, nr. 2, pp. 14-17 (in Dutch)
40. Kappen, H.J, 1996, "An overview of neural network applications," in Proceedings 6<sup>th</sup> International Congress for Computer Technology in Agriculture, (Wageningen, the Netherlands), pp. 75-79
41. Karunanithi, N, Grenney, W.J, Whitley, D and Bovee, K, 1994, *Neural networks for river flow prediction*, J. Comp. Civ. Eng., no. 8
42. Kavehercy, S, 1996, *Reinforcement learning and Artificial Neural Networks: The optimal control of water vessel process*, MSc thesis, University of Amsterdam, Amsterdam
43. Kelly, D.J *et al*, 1994, *The application of a Neural-Fuzzy controller to Process control*, Proceedings of the 1994 Biennial Conference of the North American Fuzzy Information Processing Society
44. Kolb, T, 1999, *How to work with Neural Networks in Engineering Applications*, Workshop: Neural Network in Civil Engineering, Delft
45. Krijgsman, A.J, 1993, *Artificial intelligence in Real-time control*, Proefschrift. TU Delft
46. Kusumastuti, D.I, 1999, *Application of data driven modelling techniques to rainfall-runoff in natural catchment*, MSc thesis HH370, IHE, Delft
47. Lee, H, 1997, *Optimal Reservoir Operation for Flood control Using a Hybrid Modelling Approach*, MSc thesis, IHE, Delft
48. Lee, H.K, Oh, K.D, Park, D.H, Jung, J.H, Yoon, S.J, 1997, *Fuzzy expert system to determine stream water quality classification from ecological information*, Water Science & Technology, vol. 36, no.12, pp 199-206



49. Lek., S, Delacoste, M, Baran, P, Dimopoulos, I, Lauga, J and Aulagnier, S, 1996, *Application of Neural Networks to Modelling Nonlinear Relationships in Ecology*, Ecological modelling, vol. 90, pp39-52
50. Lin, C.T, 1994, *Neural fuzzy control systems structure and parameter learning*, World Scientific Co. Ltd
51. Lint, J.W.C and Vonk, Z.C, 1999, *Neurale Netwerken Voorspellen Waterstanden*, Simposium: Neurale Netwerken in Waterbeheer, Technical University Delft, Delft (in Dutch)
52. Liu, C,L and Lin, S.C, 1998, *Artificial Neural Network forecasting for chaotic dynamic process in hydrology*, proceedings of Hydroinformatics '96 conference, pp 791-796
53. Lobbrecht, A.H, Solomatine, D.P and Bazartseren, B, 2000, *Intelligent Real-time Control in Water Resources Management*, Proc. 4<sup>th</sup> International Conference on Hydroinformatics, Iowa City, USA, Balkema, Rotterdam
54. Loke, E, Warnaar, E.A, Jacobsen, P, Nelen, F and do Ceu Almeida, M, 1997, *Artificial Neural Networks as a Tool in Urban Storm Drainage*, Water Science and Technology, vol. 36, no. 8-9, pp. 101-109
55. Maier, H.R, Dandy, G.C and Burch, M.D, 1998, *Use of artificial neural networks for modelling cyanobacteria Anabaena spp. In the river Murray, South Australia*, Ecological modelling, vol. 105, pp. 257-272
56. Maskey, S, Dibike, Y.B, Jonoski, A and Solomatine, D.P, 2000, *Groundwater Model Approximation with ANN for Selecting Optimum Pumping Strategy for Plume Removal*, Accepted for 2<sup>nd</sup> Joint Workshop on AI Methods in Civil Engineering Applications
57. Mason, J.C, Price, R.K and Tem'ne, A, 1996, *A neural network model of rainfall-runoff using radial basis functions*, Journal of Hydraulic Research, vol. 34, pp 537-548
58. Mason, J.C and Price, R.K, 1998, *Modelling sewerage system with an artificial neural network*, Austrian Journal of Statistics, vol. 27, pp. 73-82
59. Medsker, L.R, 1995, *Hybrid Intelligent Systems*, Kluwer Academic Publishers, USA
60. Minns, A.W, 1998, *Artificial Neural networks as sybsymbolic process descriptors*, PhD thesis, A.A.Balkema, Rotterdam
61. Minns, A.W and Hall, M.J, 1996, *Artificial neural networks as rainfall runoff models*, Hydrological Science Journal, vol. 41 (3)
62. Minns, A.W and Fuhrman, D.R, 2000, *Selection of input parameters for a hydrological neural network model of a snow covered catchment*, Proc. 4<sup>th</sup> International Conference on Hydroinformatics, Iowa City, USA, Balkema, Rotterdam
63. Nagel, M, 1999, *Investigation of floc structure using numerical classification methods*, Joint workshop on Neural networks in Civil Engineering, Delft
64. Nouh, M, 1996, *Simulation of water quality in sewer flows by neural networks*, proceedings of Hydroinformatics '96 conference, Balkema, Rotterdam, pp 885-891
65. Panigrahi, D.P and Mujumdar, P.P, 1997, *Application of Fuzzy logic to Reservoir Operation modelling*, National Conference on Fuzzy sets and Their Applications, Madras
66. Paruelo, J.M and Tomasel, F, 1997, *Prediction of functional characteristics of ecosystems: a comparison of artificial neural networks and regression models*, Ecological modelling, vol. 98, p. 173-186
67. Pongracz, R, Bogard, I, Duckstein, L, 1999, *Drought forecasting using atmospheric circulation and ENSO information*, Proceedings of XXVIII IAHR congress, Graz, Austria
68. Proano, O.S, 1998, *Emulation of sewer overflows using autoregressive neural network*, MSc thesis HH340, IHE, Delft, The Netherlands
69. Raman, H and Chandramouli, Y, 1996 *Deriving the general operating policy for Reservoir using neural network*, Journal of Water Resources Planning and Management, Sept/Oct pp 342-347
70. Raman, H and Sunilkumar, N, 1995, *Multivariate modelling of water resources time series using artificial neural networks*, Journal of Hydrological Sciences, vol. 40, no. 2, pp. 145-163
71. Recknagel, F, French, M, Harkonen, P and Yabunaka K.I, 1997, *Artificial neural network approach for modelling and prediction of algae blooms*, Ecological modelling, vol.96, pp. 11-28
72. Rodriguez, M.J, West, J.R, Powell, J and Serodes, J.B, 1997, *Application of two approaches to model chlorine residuals in Severn Trent Ltd (STW) distribution systems*, Water Science and Technology, vol. 36, no. 5, pp. 317-324

73. Schalkoff, R.J, 1997, *Artificial Neural Networks*, McGraw-Hill Companies, Inc,
74. Scheffer, E and MacLeod, I.M, 1999, *A predictive relational fuzzy logic model for activated sludge processes*, Proceedings, Exeter
75. Schulz, K, Huwe, B and Peiffer, S, 1999, *Parameter uncertainty in chemical equilibrium calculations using fuzzy set theory*, Journal of Hydrology, vol. 217, pp. 119-134
76. Schulze, F.H and Salverda, A.P, 1999, *Applications of Artificial Neural Networks in integrated water management: fiction or future*, Simposium: Neurale Netwerken in Waterbeheer, Technical University Delft, Delft
77. See, L and Openshaw, S, 1999, *Applying soft computing approaches to river level forecasting*, Journal of Hydrological Sciences, vol. 44, no.4, pp. 763-778
78. Sincak, P, Bundzel, M, Sokac, M, Sztruhar, D and Marsalek, J, 1998, *Urban runoff prediction by neural networks*, proceedings of Hydroinformatics '98 conference, Babovic, V & Larsen, L.C ed., Balkema, Rotterdam, pp 791-796
79. Solomatine, D.P and Avila Torres, L.A, 1996, *Neural Network Approximation of a hydrodynamic Model in Optimizing reservoir operation*, Proceedings of Hydroinformatics '96 conference, pp 201-206
80. Steyer, J.P, Rolland, D, Bouvier, J.C and Moletta, R, 1997, *Hybrid fuzzy neural network for diagnosis- application to the anaerobic treatment of wine distillery wastewater in a fluidized bed reactor*, Wat. Sci. Tech, vol. 36, no. 6-7, pp 209-217
81. Swingler, K, 1996, *Applying Neural Networks: A practical guide*, Academic Press Limited, San Diego, CA
82. Tani, T, Umamo, M, Murakoshi, S, Sato, T and Tanaka, K, 1993, *Application of Neuro-Fuzzy Hybrid control system to tank level control*, IEEE Technology Update series. Fuzzy logic and applications: Selected conference papers, pp 247-252
83. Tay, J-H and Zhang, X, 1999, *Neural Fuzzy modelling of anaerobic biological wastewater treatment plant*, Journal of Environmental Engineering, Dec, pp. 1149-1159
84. Tien Ser, T, 1995, *Development of a Real-time control system for drainage using an Object oriented approach*, MSc thesis, IHE
85. Torrence, C and Compo. G.P, 1998, *A practical guide to Wavelet analysis*, Bulletin of the American Meteorological Society, vol. 79, No. 1, pp. 61-78
86. Tsai, Y.P, Ouyang, C.F, Wu, M.Y, Chiang, W.L, *Effluent suspended solid control of activated sludge process by fuzzy control approach*, Journal of Water and Environmental Research, vol68, no. 6, pp. 1045-1053
87. Tsoukalas, L.H and Uhrig, R.E, 1997, *Fuzzy and neural approaches in engineering*, John Wiley and Sons, N.Y
88. Tzafestas, S.G and Tzafestas, C.S, 1997, *Fuzzy and neural control: Basic principles and architecture*, Methods and applications of Intelligent control: Microprocessor-based and intelligent systems engineering, Kluwer Academics, Tzafestas, S.C ed.
89. van den Boogaard, H.F.P, Ali, Md.S and Mynett, A.E, 1998, *Self-Organizing Feature maps for the analysis of hydrological and ecological data*, proceedings of Hydroinformatics '98 conference, Babovic, V & Larsen, L.C ed., Balkema, Rotterdam, pp. 733-740
90. Vapnik, V.N, 1995, *The Nature of statistical learning theory*, Springer\_Verlag, NewYork, Inc
91. Venetsky, L et al, 1996, *Real-time direct adaptive rule for fuzzy controller*, Proceedings of IEEE International Symposium on Intelligent control, pp 811-816
92. Vermeersch, L, Kerckhoffs, E and Kopelaar, H, 1999, *Neural network based structure characterization in the modelling and simulation of ill-defined systems*, Neurale Netwerken in Waterbeheer simposium paper, TU Delft
93. Vesanto, J, 1998, *SOM-based Data Visualization methods*, Intelligent Data analysis, Elsevier Science
94. Wen, C. G and Lee, C. S, 1998, *A neural network approach to multiobjective optimization for water quality in river basin*, Water resources research, vol. 34, no. 3, pp. 427-436
95. Werner, M.G.F and van den Boogaard, H.F.P, 1999, *Real-time control using an Auto Regressive Neural network*, Joint workshop on Neural networks in Civil Engineering, Delft
96. Wilson, G, 1995, *A Classifier System for the Control of an Urban Drainage Network*, MSc thesis, IHE, Delft, The Netherlands

97. Wüst, J.C, 1995, *Current prediction for Shipping Guidance*, Neural Networks: Artificial Intelligence and Industrial Applications, Kappen, B and Gielen, S (eds), Springer, Londen
98. Yan, H, 1999, *Application of data driven methods for flood forecasting*, MSc thesis HH356, IHE, Delft, The Netherlands
99. Yang, C.T and Kao, J.J, 1996, *An expert system for selecting and sequencing wastewater treatment processes*, Water Science and Technology, vol. 34, no. 3-4, p. 347-353
100. Zhang, Q and Stanley, S.J, 1997, *Forecasting raw water quality parameters for the North Saskatchewan river by neural network modelling*, Water Research, vol. 31, no. 9, pp 2340-2350
101. Zhang, Q and Stanley, S.J, February 1999, *Real Time Water Treatment Process Control with Artificial Neural Networks*, Journal of Environmental Engineering
102. Zhao, H and McAvoy, T.J, 1996, *Modelling of activated sludge water treatment processes using integrated neural networks and first principle model*, proceedings of 13<sup>th</sup> Triennial World Congress IFAC, San Francisco, pp. 455-460

## **Appendix**

**Table A.1 Review summary of ANN and FRBS application for water resources management**

Types of application in water management	Use of ANN, FRBS	Artificial Neural Networks								FRBS		Neuro-Fuzzy
		EBP	RBF	Recurrent neural networks				SOFM	PCNN	Inference	FAS	
				Time lag	Regressive	J-Elman	H-BSB					
<b>Assessment</b>												
<b>a. Resources or quantity assessment</b>												
- rainfall runoff modelling	Yes	F	F	F	F	-	-	-	-	-	-	-
- rainfall prediction	Yes	F	-	-	-	F	-	-	-	-	-	-
- urban runoff prediction	Yes	G	F	-	-	-	-	-	-	-	-	-
- drought forecasting	Yes	G/F	F	-	-	-	-	-	-	-	F	-
- groundwater flow simulation	Yes	F	-	-	-	-	-	-	-	-	-	-
- river bed evolution	Yes	F									-	-
- determining the erosion of groyne field sediment	Yes	G	-	-	-	-	-	-	-	-	-	-
- simulation of hydro morphological processes	Yes	G/F	-	-	-	-	-	-	-	-	-	-
- Q-h relationship (rating curve)	Yes	G	-	-	-	-	-	-	-	-	-	-
- Reservoir inflow prediction	Yes	G/F						-				
- Classifying discharge pattern	Yes							G				
- impact analysis	No											

Appendix

	<b>b. Water quality management</b>												
	- sewer water quality	Yes	F	-	-	-	-	-	-	-	-	-	-
	- quality of urban storm water	Yes	-	-	G/F	-	-	-	-	-	-	-	-
	- groundwater quality survey	Yes	-	-	-	-	-	-	-	-	-	-	F
	- water quality evolution in a pipe network	Yes	F	-	-	-	-	-	-	-	-	-	-
	- water quality evolution in natural flows	Yes	G	-	-	-	-	-	-	-	F	-	-
	- water quality classification of river catchment	Yes	-	-	-	-	-	-	-	-	G	-	-
	- identification of pollutant source in surface water	Yes	-	-	-	-	-	-	G/F	-	F	-	-
	- identification of groundwater contaminant	Yes	-	-	-	-	-	-	G	-	-	-	-
	<b>c. Ecological relations</b>												
	- species mass prediction	Yes	G/F	-	-	-	-	-	-	-	-	-	-
	- determining the ecoregions	Yes	-	-	-	-	-	-	-	-	-	-	-
	- classification of ecological data	Yes	-	-	-	-	-	-	G	-	F	-	-
	- ecological relations in wetland	No											
	- ecological relations in riparian zone	No											
2	<b>Planning`</b>												
	- community water demand prediction	Yes	G	-	-	-	-	-	-	-	-	-	-
	- reservoir operation planning	Yes	G/F	-	-	-	-	-	-	-	-	-	-
	- tidal water current prediction	Yes	E/G										
	- CSO simulation	Yes				G/F	G/F						
	- prediction of treatment plant characteristics	Yes										F	
	- irrigation water demand prediction	No											
3	<b>Designing</b>												
	- structure design	No	-	-	-	-	-	-	-	-	-	-	-

Appendix

4	<b>Operation and real-time control</b>												
	- optimal control of water resources system	Yes	G/F	-	-	-	-	-	-	-	-	G/F	
	- optimal reservoir system operation	Yes	G	-	-	-	-	-	-	-	G/F	-	G
	- CSO minimization	Yes	-	-	-	-	-	-	-	-	G-	-	-
	- water treatment process control	Yes	G	-	-	-	-	G	F	-	G	-	F
	- determining dosage of treatment	Yes	G/F	-	-	-	-	-	-	-	-	-	-
	- selection of treatment plant procedure	Yes	G	-	-	-	-	-	-	-	-	-	-
	- determining tidal water level	Yes		-	-	-	-	-	-	-	-	-	-
	- water bath temperature control of reactor	No											-
	- hydro power station structures	No											G
	- irrigation systems operation												
5	<b>Maintenance</b>												
	- fault detection or distinction in distribution network	Yes	F	-	F	-	-	-	-	-	G/F	-	G
	- determining leakage point in the networks	Yes	G	-	-	-	-	-	-	-	G/F	-	G
	- determining an optimal pumping rate for groundwater cleanup wells	Yes	G/F	-	-	-	-	-	-	-	-	-	-

Note: AI - Artificial Intelligence, EBP - Error BackPropagation, FRBS - Fuzzy Rule Based Systems, FAS - Fuzzy Adaptive Systems, RBF - Radial Basis Function, J-Elman - Jordan-Elman network, H-BSB - Hopfield and Brain-State-in-a-Box, SOFM - Self Organizing Feature Maps, PCNN - Principal Component Neural Networks

The performance evaluation of each technique for certain application is expressed verbally as e- Excellent, G-good, F-fair and M-moderate





**Artificial Neural Networks for Reconstruction of  
Missing Data and Runoff Forecasting:  
Application to Catchments in Salland**



**Project report**

**Part II**

By

D.P. Solomatine

A.H. Lobbrecht

Y.B. Dibike

L. Wang

**Delft**

**September, 2002**



## Contents of Part 2

<b>1 INTRODUCTION.....</b>	<b>97</b>
1.1 GENERAL .....	97
1.2 THE STUDY AREA.....	97
1.3 OBJECTIVE OF THE STUDY.....	98
<b>2 DATA PREPARATION.....</b>	<b>99</b>
2.1 DISCHARGE CALCULATION .....	100
<b>3 ARTIFICIAL NEURAL NETWORKS.....</b>	<b>103</b>
3.1 GENERAL .....	103
3.2 MULTI-LAYER PERCEPTRONS (MLPs).....	103
3.3 BACK-PROPAGATION ALGORITHM .....	104
3.4 GLOBAL VERSUS LOCAL ANNS (GANN vs LANN).....	105
<b>4 APPLICATION OF ARTIFICIAL NEURAL NETWORKS.....</b>	<b>107</b>
4.1 DATA ANALYSIS .....	107
<i>Data visualisation</i> .....	107
<i>Correlation coefficient analysis</i> .....	108
<b>5 RESULTS AND DISCUSSIONS .....</b>	<b>111</b>
5.1 APPLICATION OF ANN TO FILL MISSING DATA AT RIETBERG .....	111
5.2 APPLICATION OF ANN TO FILL MISSING DATA AT STUW 7A .....	113
5.3 APPLICATION OF LOCAL ANN MODEL TO FILL MISSING RUNOFF DATA AT STUW7A .....	115
<i>Statistical indexes</i> .....	115
5.4 APPLICATION OF ANN FOR RUNOFF FORECASTING AT RIETBERG .....	120
5.5 APPLICATION OF ANN TO RUNOFF FORECASTING AT STUW 7A .....	121
<b>6 CONCLUSION AND RECOMMENDATIONS .....</b>	<b>125</b>
<b>7 REFERENCE: .....</b>	<b>127</b>



# 1 Introduction

## 1.1 General

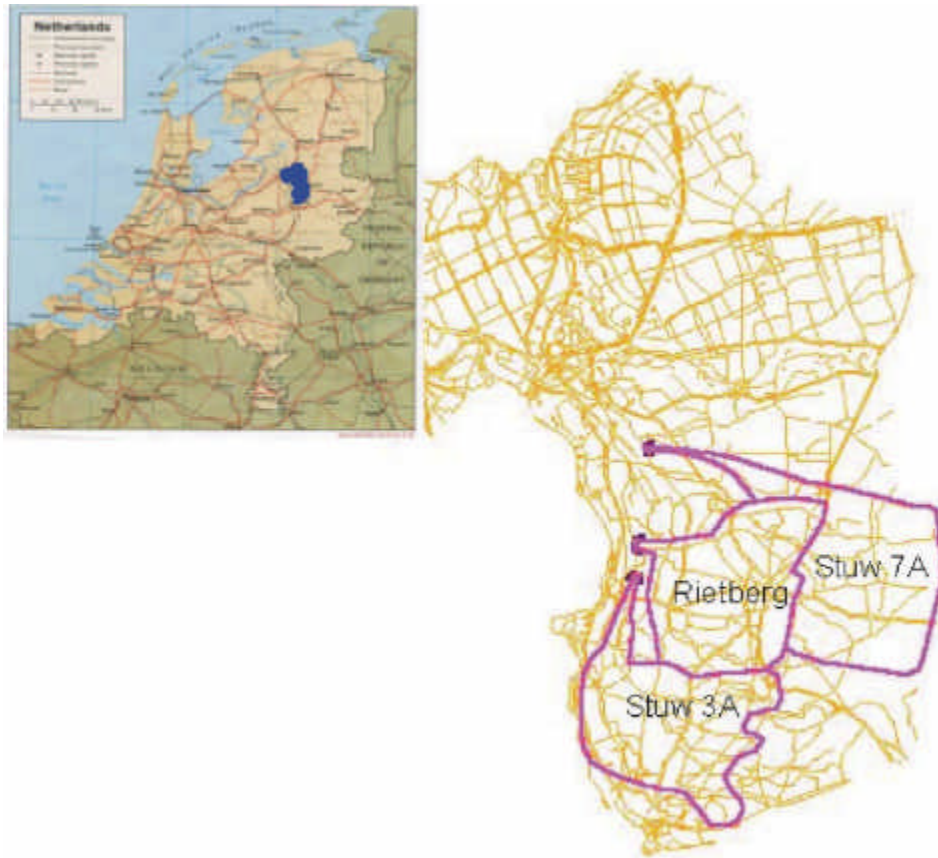
For integrated water management, monitoring of water-system behaviour is very important to understand the state of the system at any particular time, past and present, and to be able to assess the effect of various control actions on the different aspects of the water system that is to be managed. In this respect, hydrological models also play an important role in simulating water-system behaviour that could help to indicate possible trends in the living environment and nature such as flooding, drought and poor water quality.

Monitoring of data is never perfect and time series of hydrological variables often exhibits some form of deficiency due to the presence of gaps, discontinuities, and inadequate length. Such deficiencies in hydrological time series are attributable, among other, to the malfunctioning of monitoring equipment (electric or mechanical), effects of natural phenomena (e.g. earthquakes, hurricanes, landslides, etc.), problems in data transmission, storage, and retrieval processes. However, the models used for water-system analysis purposes need complete time series of historical data such as rainfall, evaporation, water level and catchment discharge, etc. for proper calibration and analysis. This urges to look for methods that permit reconstruction of missing data from what ever related information that are available so that, on the basis of complete time series, models can be build and calibrated, and can subsequently be used for prediction of the water-system behaviour.

## 1.2 The study area

The study area is in the northeastern part of the Netherlands at a place called Salland. The drainage areas at Rietberg and other two nearby catchments (as shown in Fig 1 and from now on referred by the name of the weirs at the outflow point of these areas, stuw 3A and stuw 7A) were considered at the initial stage of this study. The weir at Rietberg drains an area of 6,646 ha while stuw 3A and stuw 7A drain areas of 10,130 ha and 13,697 ha respectively. Salland is generally a gently sloping area where water management is carried out with the help of fixed weirs, controlled weirs and irrigation pumping units operated by the water board of Groot Salland.

Before 1994, two weirs of each 2.5m wide regulated the outflow from the Rietberg area. However a new weir of 3.5m long and 5m wide has replaced this since 1996. A new pumping station, which could supply water to the area during dry periods, was also added at that moment. Stuw 3A has a width of 6m while stuw 7A consists of two weirs of 2.5m wide each. In principle, these weirs are regulated either manually or automatically depending on the level of water in the canal upstream of the outlets. If the water level is high, then the weir is lowered so as to let water drain out of the area. On the other hand, if water level in the canal is low, then the weir is raised so as to keep water from leaving the area. The weirs are connected with automatic devise which keeps records of the water levels upstream and downstream of the weirs and the position of the crest of the weir itself. These records will later be used to calculate the discharge or quantity of water drained out of the drainage area.



*Figure 1 The study area at Groot Salland*

### 1.3 Objective of the study

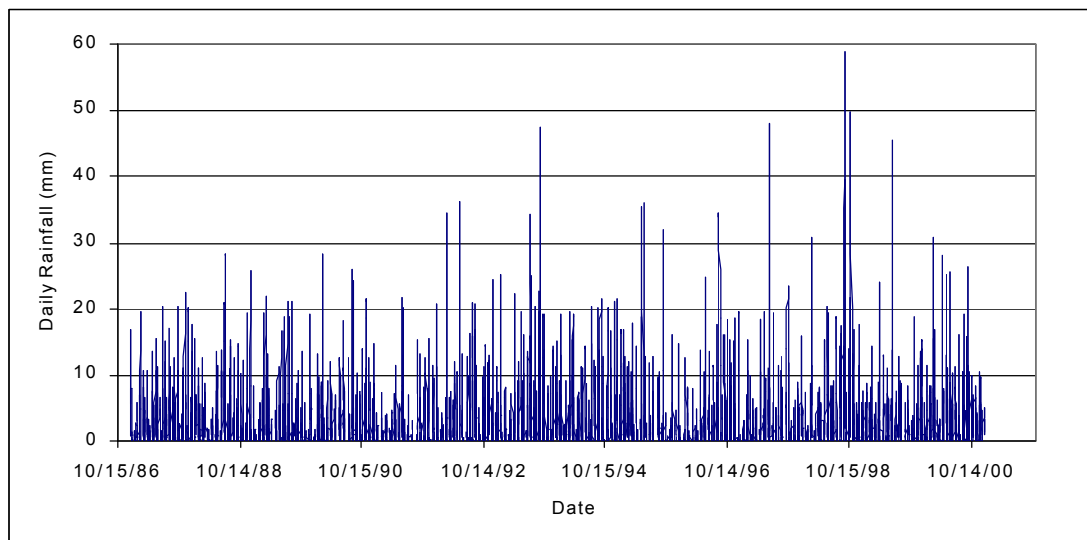
As will be shown in the next section of this report, the time series of discharges computed for each of the drainage areas has a lot of gaps. Considerable amount of data is missing due to various reasons such as reconstruction of the weir (in case of Rietberg), malfunctioning of the weirs, etc. Therefore, this research deals with the application of an artificial intelligence technique, namely that of artificial neural networks for reconstruction of missing data in historic hydrological data sets and building runoff prediction models that are based on the observed data.

## 2 Data Preparation

The first task in any modelling exercise is to collect as much observed data as possible to better understand the physical processes involved and get a better insight in to the possible relationship that might exist between the different variables that has been observed so far. The various types of data collected from the study area are listed in Table 1 and some of them are presented graphically in Figures 2 and 3.

*Table 1 Overview of available data*

	Station Name	Duration	Description
Rainfall	Heino	1987 till 2000	Daily data
Discharge or water level	Rietberg	1/18/91 till 12/20/99	Hourly data
	Stuw 3A	3/8/91 till 12/25/99	Hourly data
	Stuw 7A	3/7/91 till 8/15/99	Hourly data
Evaporation	Twente	1/1/91 till 3/31/2001	Daily data
Pumping	Rietberg	1995 till 2000	Monthly data
Groundwater Level	Approximate 90 Stations inside the research region, most of the observation taken twice a month		
Daily groundwater level	18 months of daily groundwater observation during 25, Feb. 1998 to 29, Aug. 2000 at a location with co-ordinate 220500, 494500		



*Figure 2 Daily Precipitation data at Heino*

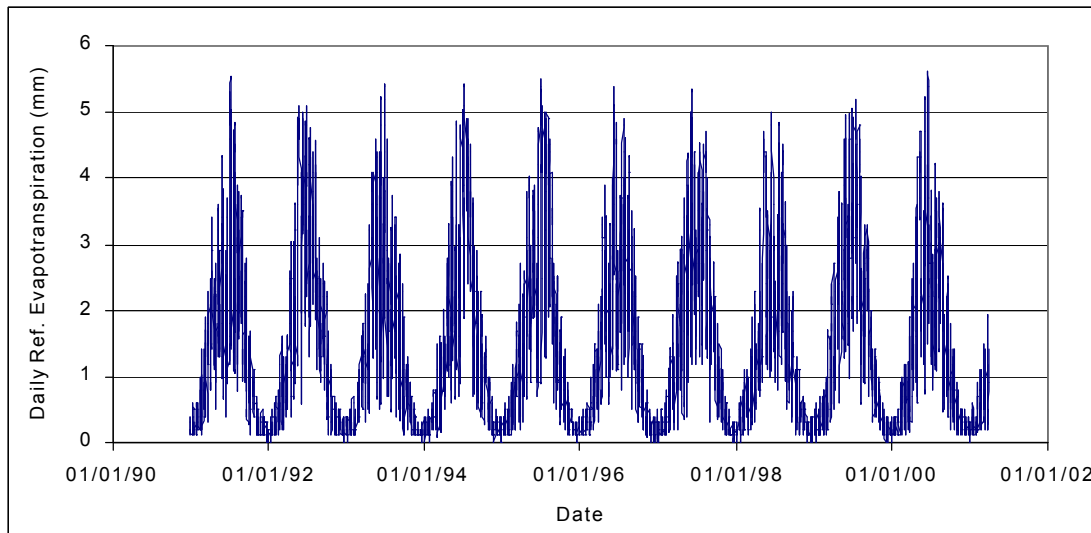


Figure 3 Daily potential evapo-transpiration data at Twente

## 2.1 Discharge calculation

As it has been mentioned earlier, the collected data includes hourly water level observations at upstream and downstream of the weir at each outlet of the three drainage areas. The hourly discharge values are therefore calculated from the observed water levels using the following weir discharge formulae (Brater et al. 1976).

- For free flow (when the level of the weir is above downstream water level)

$$Q = m * 1.7 * B * H^{3/2} \quad (1)$$

- For Submerged flow (when the level of the weir is below downstream water level)

$$\frac{Q}{Q_1} = \left[ 1 - \left( \frac{H_2}{H_1} \right)^{3/2} \right]^{0.385} \quad (2)$$

Where: B - the width of the weir

H - depth above the weir

m - weir coefficient

Q - discharge over the weir

H<sub>1</sub> - upstream water depth above the weir

H<sub>2</sub> - downstream water depth above the weir

Q<sub>1</sub> - free discharge with depth of H<sub>1</sub>

However, since the rainfall and evaporation data are observed on a daily basis, the hourly discharges calculated with the above equations are also further aggregated into daily data. This averaging of hourly discharges into daily values is also expected to reduce to some extent the influence of the regulating structures (the weirs) on the natural flow regimes at the drainage points.



The extent of the missing discharge data at Rietberg is shown in Figure 4 while the quality of the remaining data can be visualised by plotting the time series of daily discharges at the outlet of each drainage areas as shown in Figure 5.

The following important observations are made from the above plots of the time series of discharges at the outlets of the three drainage areas.

- There is a considerable difference in the relative magnitude of time series of discharge at Rietberg between the measurements taken before June 1994 and the ones taken after February 1996. This could be mainly because of the construction of a new weir and pumping station which start its operation in February 1996.
- The calculated discharges at Stuw 3A starting from November 1996 are too high and unreasonable. It was later learned that the instrument recording the level of the weir crust is malfunctioned and its records are completely erroneous.

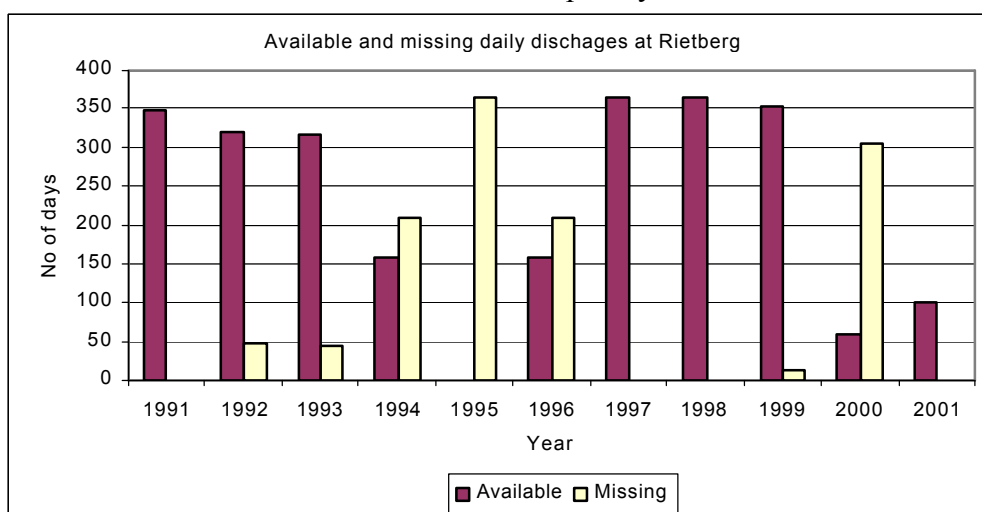


Figure 4 Available and missing data at Rietberg

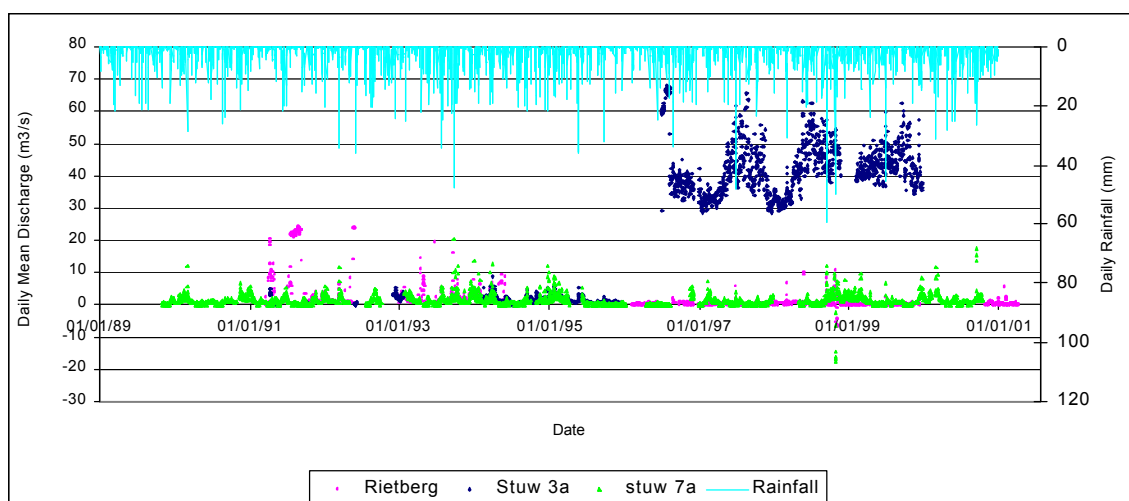


Figure 5 Time series of Rainfall data and daily discharges at the outlet of each drainage areas

- Negative discharges have been calculated both at Rietberg and Stuw 7A indicating water flowing from downstream up into the respective drainage areas. A closer investigation revealed that these phenomena have indeed been witnessed during the flood events of October/November 1998.

Because of this and other considerations, it was decided to discard the data at Stuw 3A and consider only the data at Rietberg and Stuw 7A for further analysis and modelling. It was also found more appropriate to consider the time series of discharge at Rietberg that was collected after the construction of the new weir. The periods with negative discharge values are also removed from the time series before any further analysis.

### 3 Artificial Neural Networks

#### 3.1 General

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones, and this is also true for the case of ANNs as well.

Neural networks, with their remarkable ability to derive a non linear relationship from imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an “expert” in the category of information it has been given to analyse. This “expert” can then be used to provide projections given new situations of interest and answer “what if” questions. Therefore, generally speaking, the neural networks do not need much of a detailed description or formulation of the underlying process. Rather, it follows the method of adaptive learning, an ability to learn how to do tasks based on the data given for training or initial experience by creating its own organisation or representation of the information it receives during learning time.

#### 3.2 Multi-layer perceptrons (MLPs)

MLPs is a kind of Feed-forward ANNs which only allow signals to travel one way only from input to output. It is the commonest type of artificial neural network consisting of three or more layers of units which are commonly called processing elements (PEs): a layer of “input” units is connected to a layer of “hidden” units, which is connected to a layer of “output” units (see Figure 6). There is no feedback (loops) i.e. the output of any one layer does not affect that same layer or any previous layers. Therefore, MLP networks are the simplest types of ANNs and they are extensively used in pattern recognition and most of the application in water resources resort to this type of ANN (Zealand et al., 1999; Panu, et al. 2000; Robert, et al. 2000).

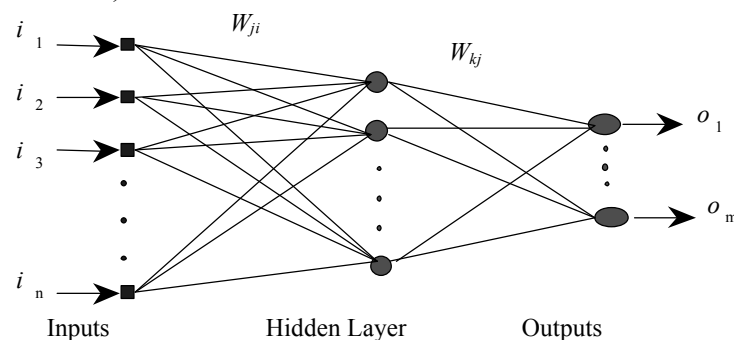


Figure 6 A typical structure of a feed-forward multi-layered perceptron

The activity of the input units represents the raw information that is fed into the network. The unique role of input layer is offering an interface to the information outside the system while the function of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. Similarly, the behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units. The weights of the connections between the units in the input and hidden layer or between the units in the hidden and output layer determine when each unit is going to be active, and so by modifying these weights, an MLP network can determine its representation. Each hidden or output unit is consisted of two parts: one is a summation and the other is an activation function. Figure 7 is the schematic of such kind of node.

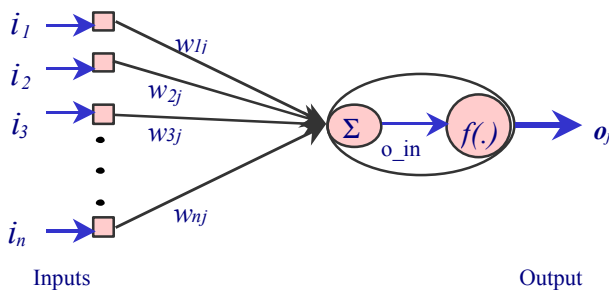


Figure 7 A schematic illustration of a neurone

### 3.3 Back-propagation algorithm

In order to train a neural network to perform some task, the weights of each connection between every units must be adjusted in such a way that the difference between the desired output and the actual output from the network is reduced and gradually converge so that the final network outputs become as close to the observed output values as possible. This process requires that the neural network compute the error derivative of the weights. In other words, it must calculate how the error changes as each weight is increased or decreased slightly.

The back propagation algorithm is the most widely used method for determining the error derivative of the weights. Valluru and Hayagriva (1993, cited by Zealand, 1999) estimate that over 80% of all neural network application use the BP learning algorithm. The back-propagation algorithm computes each error derivative of the weights by first computing the rate at which the error changes as the activity level of a unit is changed. Considering a network with three layers as shown in Fig.6, at the completion of a pass through the entire data set, all the nodes change their weights based on the accumulated derivatives of the error with respect to each weight and these changes moves the weights in the direction in which the error declines most quickly.

If we let  $w_m$  represent the value after iteration  $m$  of a weight  $w$ , which may be either a hidden-node weight  $w_{ij}$  or an output node weight  $w_{jk}$ , then:

$$w_m = w_{m-1} + \Delta w_m \quad (3)$$

where  $\Delta w_m$  is the change in the weight  $w$  at the end of iteration  $m$  and is calculated as:

$$\Delta w_m = -e d_m \quad (4)$$

where  $\varepsilon$  is the parameter controlling the proportion by which the weights are changed. The user sets the value of this parameter and the term  $d_m$  is given by:

$$d_m = \sum_{n=1}^N \left( \frac{\partial E}{\partial w_m} \right)_n \quad (5)$$

where  $N$  is the total number of examples, and  $E$  is the simulation output error.

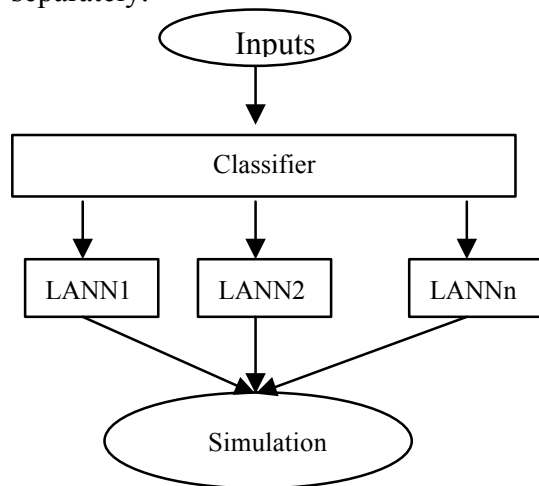
The performance of this algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. There are no standard methods to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

### 3.4 Global versus local ANNs (GANN vs LANN)

In this study, global neural network (GANN) refers to cases where a single ANN is used to model all the available input-output data set in a single architecture, where as local neural network (LANN) refers to cases where an ANN is used to model each distinct part of the data set. Because of its global nature, GANN can not capture specific information which are mainly related to specific parts of the data set. As a result, only part of the data, which appear more frequently, will be simulated more accurately. The overlap and contradictories existing between the different segments of the data set may also “confuse” the GANN and make the training processes a difficult and time-consuming task. One of the important properties of ANN is its ability to generalise complex non-linear relationship from the training data. However, due to its ability of generalisation, some special features in the data may be treated as noise and filtered to avoid overfitting. In other words, if one tries to use GANN to model the whole data set, the model will only offer the so-called best compromise solution to satisfy the generalisation feature. Moreover, if the training data set is very large, then the training process is very slow as the network attempts to learn all aspects of the input space in a global fashion and achieve the best compromising solution to represent all the cases.

If we consider rainfall-runoff processes, the low flow is usually dominated by initial loss while the high flow is mainly affected by rainfall intensity (Zhang et al. 2000). But, when ANN is applied to model such problems where there are different hydrological processes involved in generating the flow at the different parts of the hydrograph, a single ANN (GANN) usually simulates median events better than extreme (high or low flow) events. So before using ANNs to capture the rainfall-runoff relationships, it is better to classify the hydrologic sequences into several homogeneously grouped data sets which represent separate part of the hydrograph and train local ANN models to map the input and output relationship in the range of each individual data set. This will lead to designing several local ANN (LANN) models for each group of events where each LANN could have different input patterns to contain the main influencing factors. For example, rainfall and previous precipitation data will be the main components in the input layer of LANN intended to simulate wet periods while for dry periods, the LANN’s input parameters should mainly consist of evaporation. Under such situations, a classifier could be introduced before the data is fed into the ANN models (see Figure 8). Then several local ANNs are trained using the

clustered data sets and for each LANN the optimal input pattern will be determined separately.



*Figure 8 The illustration of a classifier based local ANN model*

## 4 Application of Artificial Neural Networks

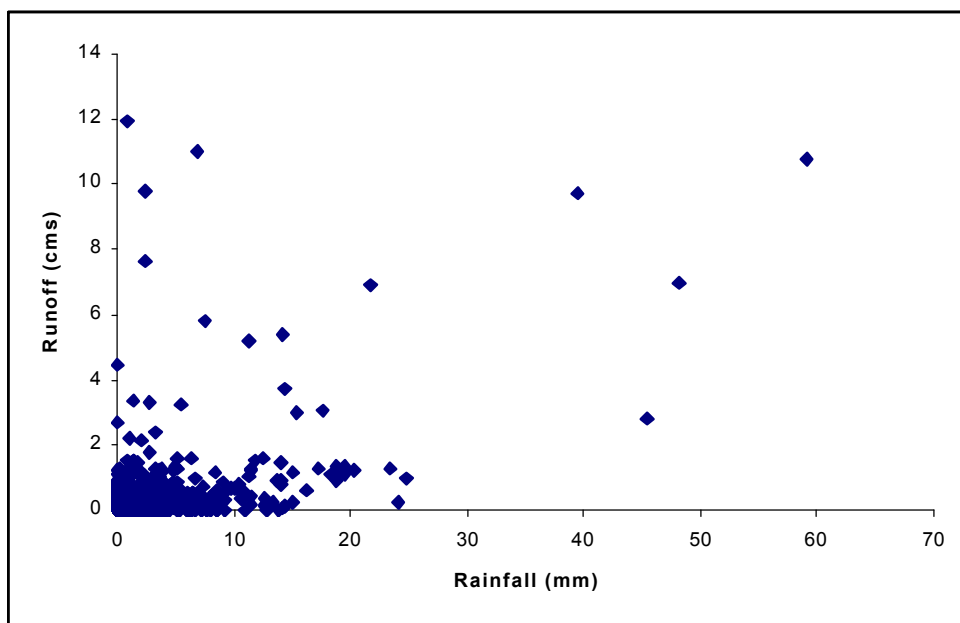
Application of data-driven modelling techniques such as ANNs in hydrology and water resources modelling is becoming more popular in recent years. They have been applied successfully for rainfall-runoff modelling, stream flow forecasting and other similar hydrologic problems. In the following section, the applicability of ANNs for reconstruction of missing discharge data and runoff forecasting at Rietberg and Stuw 7A is investigated.

### 4.1 Data analysis

There are a number of conditions required for the appropriate selection of hydrological time series data. Such conditions included availability of uninterrupted discharge data with suitable lengths, availability of discharge data from nearby drainage areas that are subject to similar physiographical properties, etc. (Khalil et al, 2001).

#### *Data visualisation*

Data visualisation and identification of outliers are some of the primary tasks in data analysis for model construction. Graphs could be used to visualise the data and offer some insight into the inner structure of variables. Visualising multiple data with the help of scatter plots is one way of revealing possible relationships between the different variables. For example, the scatter plot between rainfall and runoff at Rietberg (Figure 9) shows an overall trend of positive correlation between the two variables. But it is also easy to observe that the rainfall-runoff relationship is more complex than a one-to-one correspondence between these variables. Instead, there seems to be highly non-linear relationship between the two variables. In Figure 10, the scatter plot of Stuw7A shows a more visible relationship between the rainfall and runoff at this reference site.



*Figure 9 Scatter Plot of Rainfall and Discharge of Rietberg*

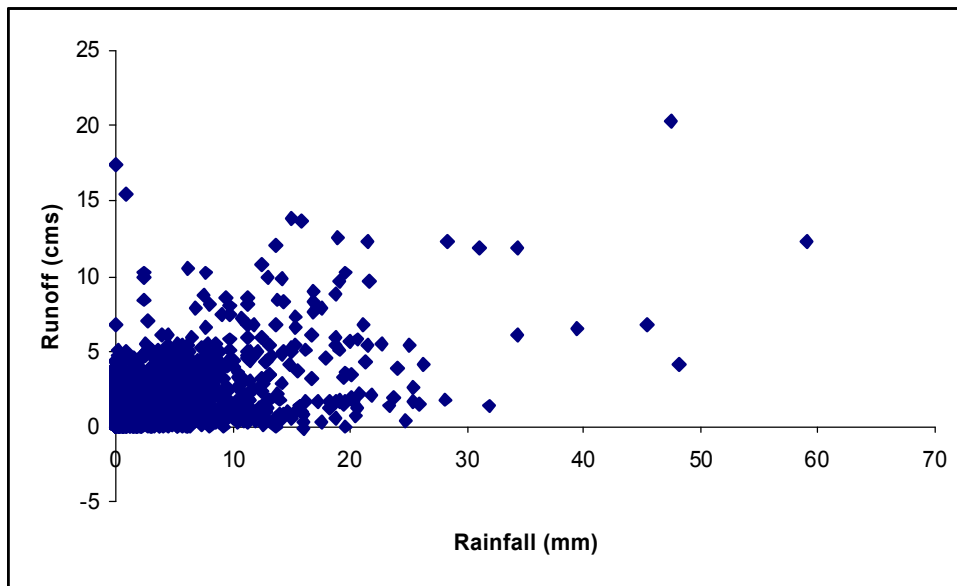


Figure 10 Scatter Plot of Rainfall and Discharge of Stuw7A

### ***Correlation coefficient analysis***

Linear correlation coefficients can be calculated between concurrent measurements of different variables or between time lagged values of the same variable or time lagged values of different variables. However, correlation analysis is not only offering benefits to linear application, but also could be used as a way to reveal the structure of a non-linear system. The time lag vectors have a strong connection to the internal dynamics of the complex system and can be used to detect the non-linearity of the time series and guide the construction of optimal neural network models' architecture.

In this study, rainfall and evaporation were treated as main influence variables since they are among the two principal daily factors influencing discharge (water levels). Moreover, the amount of discharge in a day also depends on precipitation that occurred in previous days. Therefore, to determine the extent to which the antecedent input values affect the present output, statistical correlation analysis were performed and the results are provided in Figures 11 and 12. These results are used as a reference while designing the proper architecture of Artificial Neural Networks models.



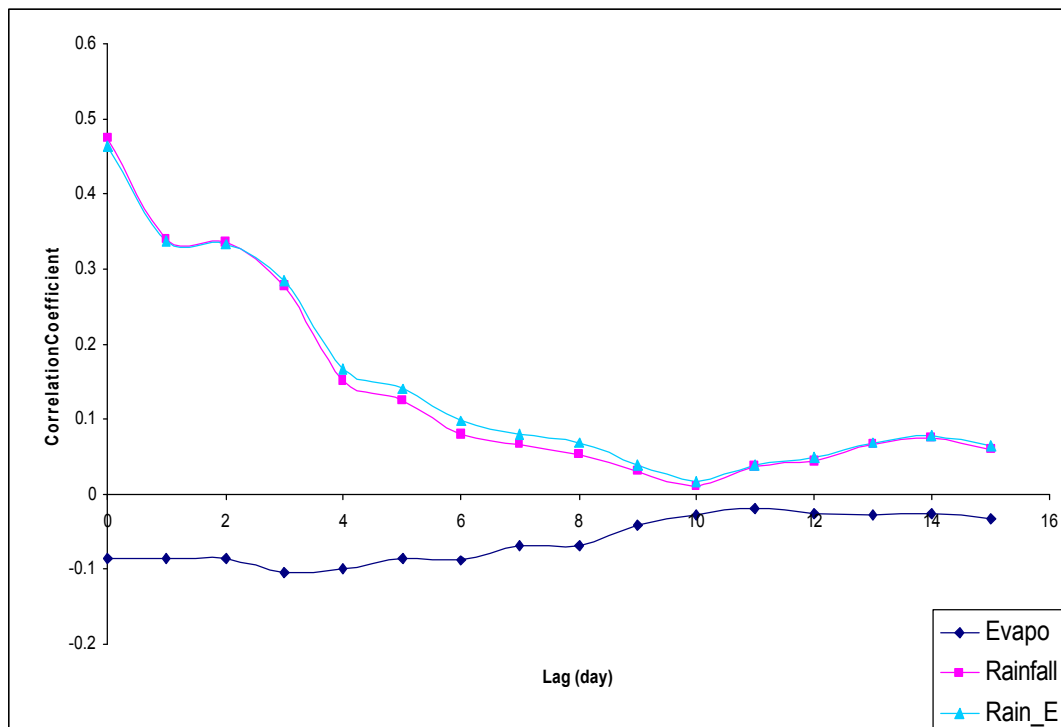


Figure 11 Correlation Coefficients of different variables with discharge of Rietberg

Figure 11 shows that the highest correlation between discharge and rainfall occurs for the concurrent events (lag of zero days), that is to say, the current rainfall influences runoff significantly and the correlation weakens with the increase in the lag time. However, significant correlation can still be observed with lag time of one to three days. This point could also be supported by the fact the moving average value of rainfall with the window of 4 days is highly correlated with discharge. So, while setting up a data-driven model, at least rainfall data of the past three days should be considered.

The concurrent and all lagged evaporation data show negatively correlated with discharge. The values are also rather small compare with the correlation coefficient between rainfall and discharge in an absolute scale. As a result, the net rainfall (rainfall minus evaporation) affects discharge almost the same way with rainfall from correlation point of view. While the values of the coefficient in Figure 11 are all below 0.5, trends could be obscure even when a large number of measurements is available (Allan P. 1999). So a number of numerical tests should be performed to find the proper input patterns in the later phase of ANN model building.

Fore the case of Stuw 7A, Figure 12 shows that the correlation between lagged evaporation and the corresponding discharges are more significant. Though the highest correlation is obtained for the concurrent values, antecedent rainfall and evaporations of up to five days lag are found to be significant. Moreover, the net rainfall (rainfall minus evaporation) with different lags show higher correlation with discharge than rainfall alone.

Figure 13 also shows that the correlation coefficient with 3 days lagged moving average of rainfall or rainfall minus evaporation are highly significant, indicating that the cumulative effects of the rainfall of current and past two day is the significant influence factors and contribute significantly to runoff.

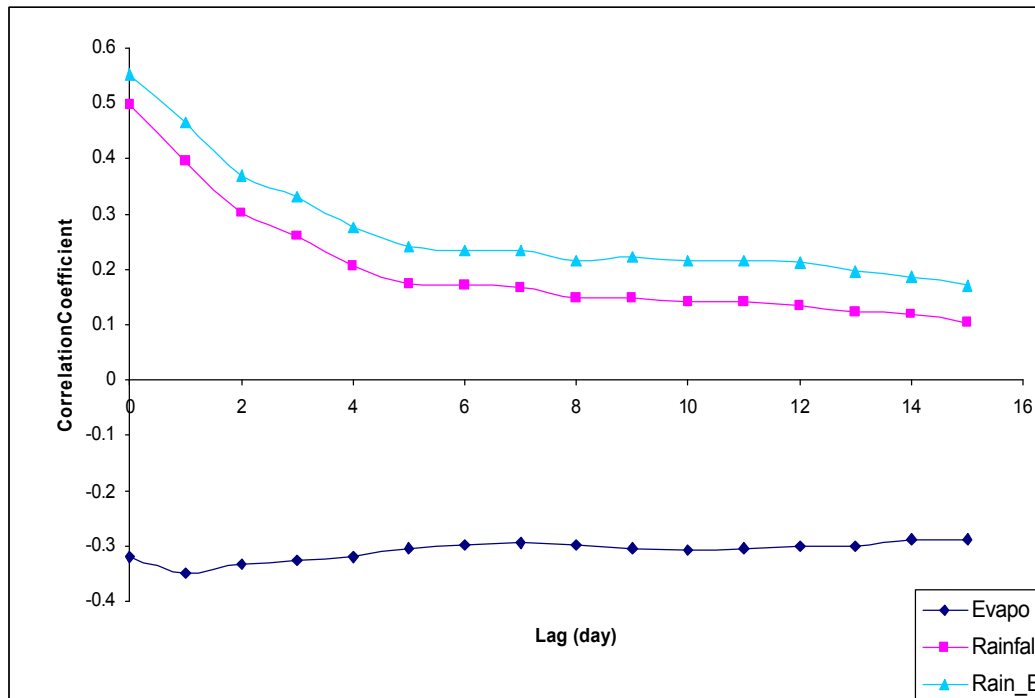


Figure 12 Correlation Coefficient of different variables with discharge of Stuw 7A

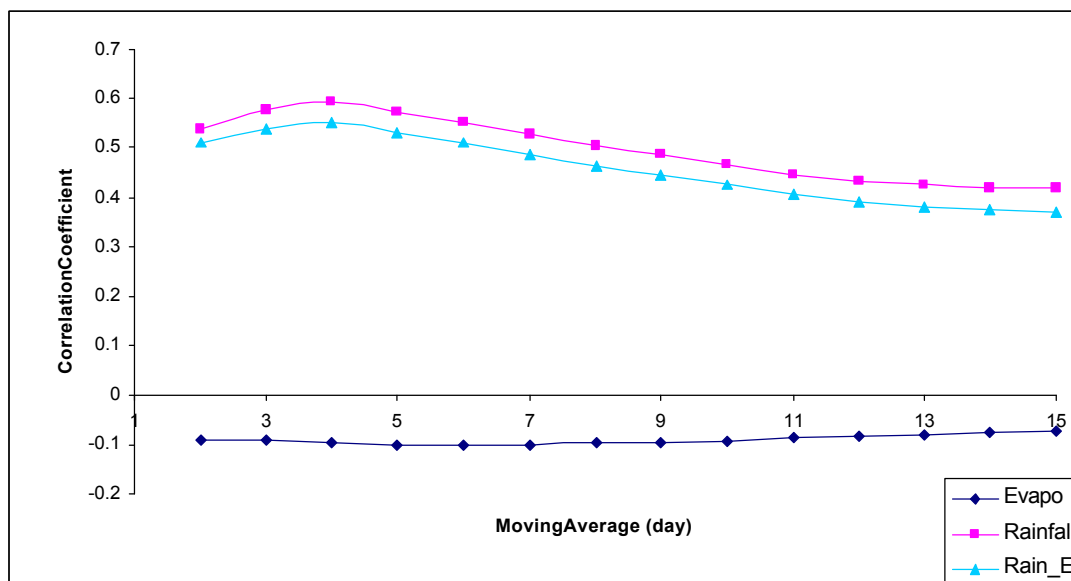


Figure 13 Correlation Coefficient Analysis of moving average values of different variables and discharge of Rietberg

Based on the above correlation analysis between rainfall (evapotranspiration, rainfall minus evaporation) and discharge, different schemes are designed, calibrated and verified to identify the optimal neural network architecture.

## 5 Results and Discussions

### 5.1 Application of ANN to fill missing data at Rietberg

To construct ANN models which can be used for the task of filling missing values in the discharge time series at Rietberg, the daily data of 1996 to 1998 (with a total of 671 exemplars) were used for training, while the daily data of 1999 (with a total of 346 exemplars) were used for testing the performance of the network on a data set which was not considered during the training. A large number of ANN models were constructed to find the optimal network architecture, and after a through investigation, eight different schemes of MLP networks with different combinations of input parameters were finally selected as best performing architectures. These schemes and the corresponding input parameters (e.g. input patterns) along with the optimal hidden nodes are listed in Table 2. Three numerical performance criterion, namely Normalised Mean Square Error (NMSE), Mean Square Error (MSE) and Correlation Coefficient (r) between the simulated and observe outputs were used to judge the performance of each schemes and identify the most optimal structures.

Table 2 The performance of 8 schemes of Rietberg of data infilling

Input patterns	Hidden nodes	NMSE		MSE		r	
		Training	Testing	Training	Testing	Training	Testing
$P_{ma4}$	2	0.46	0.899	0.588	0.206	0.734	0.476
$P_{ma4}, P_t$	3	0.454	0.853	0.580	0.196	0.739	0.511
$P_t$	3	0.690	1.43	0.882	0.329	0.557	0.384
$P_t, P_{t-1}$	5	0.570	1.424	0.729	0.326	0.656	0.436
$P_t, P_{t-1}, P_{t-2}, E_t, E_{t-1}$	5	0.508	0.725	0.649	0.166	0.701	0.629
	5(without outliers)	0.317	<b>0.648</b>	0.291	<b>0.149</b>	0.827	<b>0.643</b>
$P_t, P_{t-1}, P_{t-2}, P_{t-3}, P_{t-4}, P_{t-5}$	4	0.454	0.814	0.580	0.187	0.739	0.516
$P_t, P_{t-1}, P_{t-2}, P_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}$	10(without outliers)	0.226	0.655	0.208	0.150	0.880	0.622
$P_t, P_{t-1}, P_{t-2}, P_{t-3}, Q_{7A,t-3}, Q_{7A,t-4}, Q_{7A,t-5}$	8(without outliers)	0.098	0.974	0.133	0.224	0.950	0.579
	8(Monitor)	0.126	0.814	0.170	0.187	0.935	0.567

The optimal structure can be identified from the above table and be expressed as following:

$$Q_t = f(P_t, P_{t-1}, P_{t-2}, E_t, E_{t-1}) \quad (6)$$

where  $Q_t$  is the runoff for the current day (t);

$P_t$  is the concurrent rainfall;

$E_t$  is the concurrent evaporation;

$Q_{7A,t-i}$  is the discharge of Stuw7A with  $i$  days lag.

The value (0, 1 or 2) in the sub-index of input patterns indicates the relevant antecedent values that have a significant influence on the runoff value of the current record as it has been estimated using cross-correlation between the dependant and independent variables as described earlier. Rainfall values from previous records were needed as network inputs because the lag time of watershed response was extending up to 3 days. The graphical presentation of the results of the optimal network is shown in Figure 14.

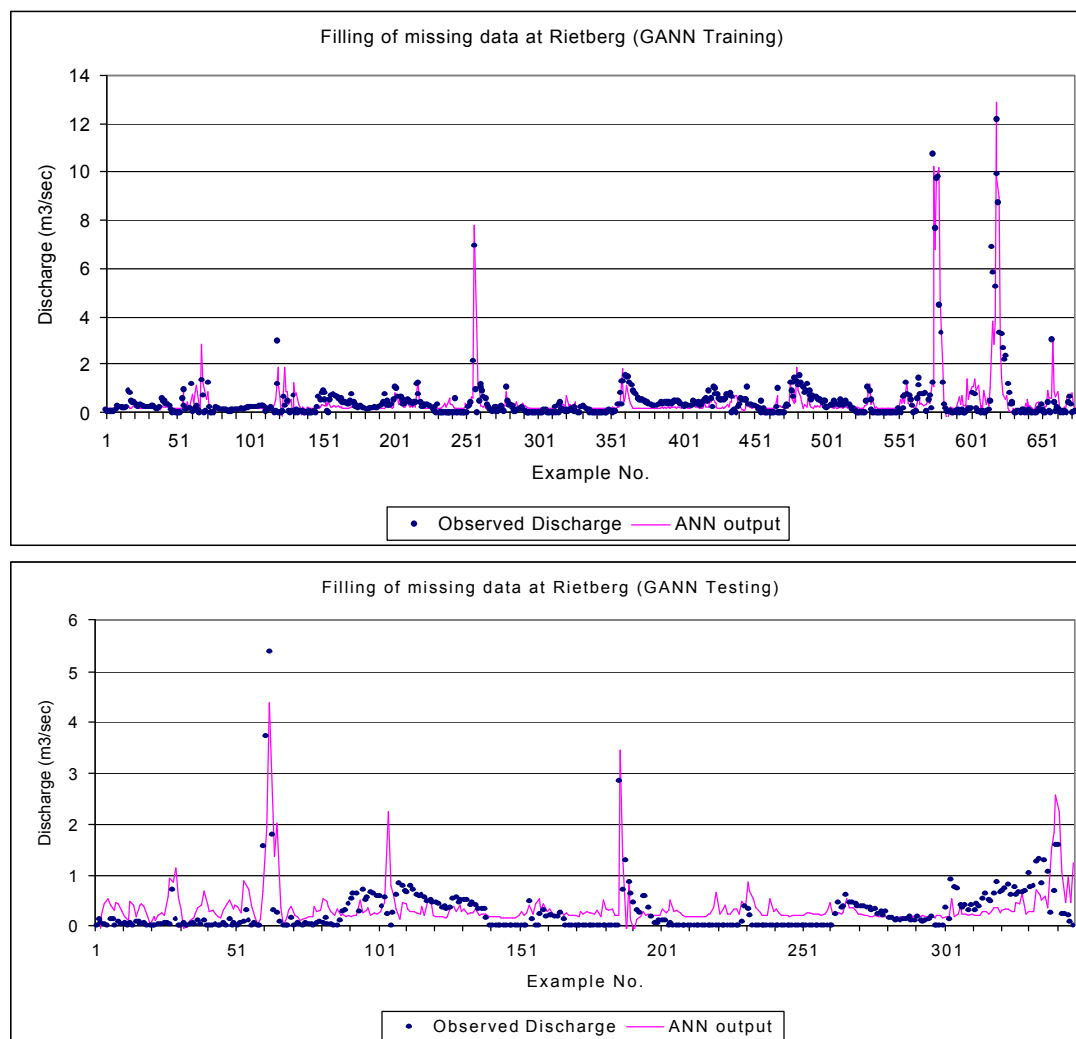


Figure 14. Comparisons of measured and calculated runoff of the optimal model (upper for training; lower for testing)

The graph in Figure 14 shows that the calculated and measured discharges on the test data set fit reasonable well. However, one can not claim this to be a perfect fit. One possible reason for this could be the lack of enough measured data which cover the whole range of possible rainfall and discharge values. The other reason could be the fact that the drainage area (the polder) is highly regulated by the weir at Rietberg and other small structures inside the area. Even though aggregating both the rainfall and discharge values on a daily basis could minimise the influence of the regulating structures, this may not be enough to completely remove the effect this could have on the rainfall – discharge relationship. Therefore, the fact

that the effect of this human intervention is not included in the models might be yet another factor for the modest performance of the ANN models.

However, it is still possible to use the identified optimal neural network to carry out the data infilling work under the assumption that the generated series still keep the main statistical features of the observed series.

## 5.2 Application of ANN to fill missing data at Stuw 7A

Compared to Rietberg, the data from Stuw7A has less missing values and also seems more consistent. Nevertheless, similar ANN modelling experiments to that of Rietberg are performed to find the relationship between the different hydrological variables of the drainage area with the discharge at the regulating structures of Stuw7A. Since we have relatively long time series at this location, the data set is divided into two groups: from 1989 to 1997 (a total of 2398 exemplars) used for training and testing was done with the data from 1999 to 2000 (a total of 638 exemplars). The different schemes with different input parameters investigated are listed in Table 3. The input patterns for each scheme are formed mainly according to cross correlation analysis. Due to the different features illustrated by such analysis, the selected input patterns are a little bit different from that of Rietberg. In this basin, evaporation has been found to have stronger influence to runoff than that of Rietberg.

Table 3 The performance of the GANN models

Input patterns	Hidden nodes	NMSE		MSE		r	
		Training	Testing	Training	Testing	Training	Testing
$PE_{ma4}$	4	0.512	0.590	1.202	1.857	0.699	0.651
$PE_{ma4}, PE_{ma4,t-1}$	5	0.464	0.599	1.088	1.866	0.732	0.648
$PE_{t-1}, PE_t$	6	0.512	0.643	1.204	2.003	0.698	0.616
$PE_{t-3}, PE_{t-2}, PE_{t-1}, PE_t$	7	0.442	0.586	1.039	1.825	0.747	0.664
$PE_{t-5}, PE_{t-4}, PE_{t-3}, PE_{t-2}, PE_{t-1}, PE_t$	9	0.625	0.534	0.941	1.663	0.774	0.698
	8(without outliers)	<b>0.402</b>	<b>0.416</b>	<b>0.945</b>	<b>1.016</b>	<b>0.773</b>	<b>0.778</b>
$P_t, P_{t-1}, P_{t-2}, E_t, E_{t-1}$	7	0.423	0.587	0.993	1.620	0.760	0.662
$P_t, P_{t-1}, P_{t-2}, P_{t-3}, P_{t-4}, P_{t-5}$	5	0.548	0.735	1.205	2.029	0.672	0.522
$P_t, P_{t-1}, P_{t-2}, P_{t-3}, E_t, E_{t-1}, E_{t-2}, E_{t-3}$	11	0.395	0.586	0.928	1.616	0.779	0.666

The optimal structure can be identified from the above table and be expressed as follows:

$$Q_t = f(PE_t, PE_{t-1}, PE_{t-2}, PE_{t-3}, PE_{t-4}, PE_{t-5}) \quad (7)$$

where:  $Q_t$  is the runoff for the current day (t);

$PE_t$  is the concurrent rainfall minus evaporation;

$PE_{t-1}$  is the rainfall minus evaporation with lag 1.

The sub-index values of 0 to 5 in of input patterns indicates the relevance of time lagged value up to 5 days in the past as they have a significant influence on the concurrent runoff values as estimated using cross-correlation between the dependant and independent variables as described earlier. Rainfall minus evaporation values is the dominant factor that influence runoff as illustrated by the former correlation analysis. Graphical presentation of the simulation results of the optimal network is shown in Figure 15.

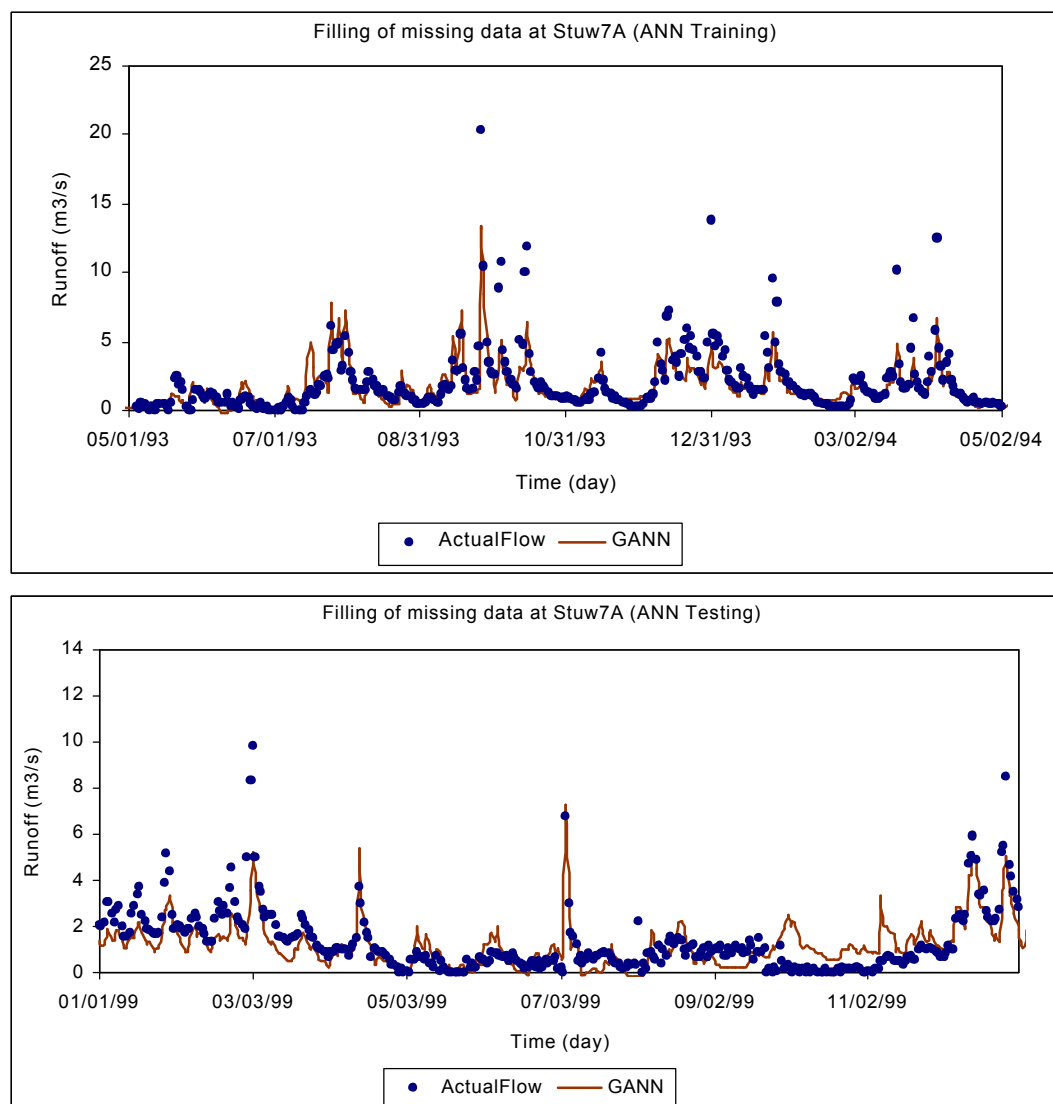


Figure 15. The comparisons of measured and calculated runoff of Stuw7A

Both Table 3 and Figure 15, shows that ANN models perform slightly better in the case of Stuw7A than that of Rietberg. However, these results are still far from being perfect. This could once again be attributed to the same reasons given previously for the case of Rietberg, such as the quality of the data and influence of human intervention corresponding to the operation of regulating structures. Nevertheless, a closer look at figure 15 shows that for low flows, the performance is quiet better than that of high flows. That will serve as a motivation to construct more local ANN models instead of one unique singular ANN model to represent the rainfall-runoff process over the entire range of the data.

### 5.3 Application of Local ANN model to fill missing runoff data at Stuw7A

The performance of data - driven models is significantly affected by the proper selection of model parameters. For hydrological modelling, segment (local) calibration procedures that are capable of simultaneously considering the different portions of runoff generation mechanism should be developed. Therefore, this investigation present a classifier based ANN (LANN) model for continuous runoff modelling to overcome the shortcoming of global ANN models which may experience major problems in capturing all the different components of the runoff generation process.

In case of LANN, all the available data set will be grouped into smaller similar data group before constructing ANN models. So several local ANN models will be trained instead of unique ANN model in order to cover the different flow regimes that might exist in the system. Each LANN will have different input patterns that would contain the main influencing factors. For LANN related to wet periods, rainfall and previous precipitation will be the main components of input space. For dry periods, the LANN's input parameters may consist mainly of evaporation.

The effectiveness and relative advantages of LANN is investigated by applying the method for modelling the rainfall-runoff processes at Stuw7A. Unlike the case of Rietberg (with 3 to 4 years data set), Stuw7A has longer data series (about 10 years) and it is more convenient to apply LANN on the later than the former in order to make reasonable comparison of the methods. To assess the relative performance of LANN approach, NN schemes with the same architecture and input variables to that of GANN models are constructed. For continuous simulation of runoff, the samples should contain base, low, medium and high flow events simultaneously and continuously and the separations of low flow period average flow period and high flow period should be performed. For the convenience of carry out forecasting, the flow separation in this particular study is done on monthly bases.

A classification of hydrograph "magnitude" could be derived through clustering analysis of bulk flow indices estimated from the discharge observations for each month. Therefore, instead of grouping the data set into subset using the original data sorted by month, another approach is applied which first subtracted the flow index, then cluster the 12 months into different groups using these statistical values. Since there are lot of missing values in the data, the months which lost more than 10 days of data are removed from further clustering analysis. For the remaining months, bulk flow indices were quantified. (Hannah, et al. 2000).

#### *Statistical indexes*

A classification of hydrograph "magnitude" could be derived through the cluster analysis of flow indices estimated from discharge observations for each month. The indices with the hydrological justification for their inclusion are listed in Table 4. (Hannah, et al. 2000).

*Table 4. Flow Indexes and their hydrological meaning*

Indices	Name and Formula	Hydrological Meaning
$Q_{\text{mean}}$	Mean monthly discharge ( $\text{m}^3/\text{s}$ )	The mean of the discharge sequence.
$Q_{\text{max}}$	Maximum monthly peak ( $\text{m}^3/\text{s}$ )	The maximum value of discharge.
$Q_{\text{min}}$	Minimum monthly discharge ( $\text{m}^3/\text{s}$ )	Baseflow discharge

$Q_{\text{range}}$	$Q_{\text{max}} - Q_{\text{min}}$	Represents the maximum amplitude of the monthly flow cycle
$Q_{\text{std}}$	The monthly standard deviation of observed discharge	Provides the monthly variability in discharge around mean

*Table 5 Indexes of statistical features of 12 months*

	$Q_{\text{mean}}$	$Q_{\text{median}}$	$Q_{\text{std}}$	$Q_{\text{range}}$	$Q_{\text{min}}$	$Q_{\text{max}}$
Jan	2.049	1.610	1.706	9.640	0	9.64
Feb	2.024	1.930	1.227	8.120	0.2	8.32
Mar	2.016	1.545	1.798	12.290	0	12.29
Apr	0.845	0.660	1.076	12.520	0	12.52
May	0.563	0.430	0.539	2.480	0	2.48
Jun	0.894	0.665	0.851	5.580	0	5.58
Jul	0.758	0.545	0.882	6.720	0	6.72
Aug	0.676	0.520	0.622	4.330	-0.13	4.2
Sep	1.629	1.090	2.397	20.360	0	20.36
Oct	1.291	0.690	1.886	11.980	0	11.98
Nov	1.301	0.780	1.409	7.550	0	7.55
Dec	2.098	1.680	1.913	13.710	0.05	13.76

*Table 6. Groups formed based on statistical indexes*

Group	Group 1	Group 2	Group 3	Group 4
Components	Dec., Jan., Feb.	Mar., Apr., May	Jun., Jul., Aug	Sep., Oct., Nov.
LANN label	LANN1	LANN2	LANN3	LANN4
No. of exemplars	785	802	932	792

The basic indices are drawn monthly and listed in Table 5. Tree clustering module of STATISTICS is used to carry out the clustering task and then a classifier is formed according to the results in Table 6. For each class, a separate ANN model is constructed.

During wet period, rainfall is the dominant factor and is the significant variable. However, during dry period, evaporation contributes high weight and forms the main part of input variables. To confirm these assumptions, the cross – correlation analysis between variables of different lag time in each group and the corresponding runoff are performed and the results are listed in Table 7.



Table 7. Correlation analysis between variables of different lag time in each group and the corresponding runoff.

	Lags	0	1	2	3	4	5
All Data	Evaporation	-0.321	-.349	-0.333	-0.326	-0.321	-0.305
	Rainfall	0.499	0.396	0.301	0.262	0.206	0.172
	Rain – E	0.554	0.465	0.370	0.329	0.276	0.242
Group1	Evaporation	-0.023	-0.147	-0.121	-0.104	-0.116	-0.102
	Rainfall	<b>0.672</b>	0.548	0.435	0.361	0.309	0.278
	Rain – E	<b>0.701</b>	0.566	0.444	0.371	0.317	0.289
Group2	Evaporation	-0.382	<b>-0.431</b>	-0.405	-0.381	0.373	-0.342
	Rainfall	<b>0.571</b>	0.405	0.286	0.224	0.168	0.120
	Rain – E	<b>0.603</b>	0.484	0.369	0.294	0.250	0.198
Group3	Evaporation	-0.271	-0.315	-0.286	-0.282	-0.259	-0.226
	Rainfall	0.485	0.370	0.340	0.287	0.258	0.224
	Rain – E	0.513	0.410	0.373	0.324	0.292	0.253
Group4	Evaporation	-0.055	-0.118	-0.074	-0.058	-0.052	-0.015
	Rainfall	<b>0.520</b>	0.439	0.311	0.292	0.203	0.163
	Rain – E	0.522	0.447	0.310	0.287	0.197	0.158

For Group1 (formed by the data of Dec., Jan., Feb.) and Group4 (formed by the data of Sep., Oct., Nov.), the correlation between evaporation and runoff become very small, so in the following models design stage, this fact should be considered and less variables from evaporation should be put into the input patterns for these two groups. For Group2 (formed by the data of Mar., Apr., May), the correlation of E and runoff become larger than that of whole data set with runoff, and the input parameters of this group should pay a little bit attention to evaporation.

For each group, two ANN schemes were considered. One scheme only uses net rainfall (precipitation minus evaporation); another schemes contain more variables related to evaporation than rainfall related. Table 8 listed the related input variables, the optimal hidden nodes and the performance. Two performance criteria, namely, NMSE and correlation coefficient (r) are used to measure the LANN models' performance.

Table 8 The performance of 2 LANN schemes

Models	Input Patterns	Hidden nodes	NMSE		r	
			Training	Testing	Training	Testing
GANN	$PE_t, PE_{t-1}$	9	0.625	0.534	0.774	0.698
LANN1_1	$PE_{t-2}, PE_{t-3}$	8	<b>0.200</b>	<b>0.381</b>	<b>0.894</b>	<b>0.889</b>
LANN2_1	$PE_{t-4}, PE_{t-5}$	7	0.202	0.474	0.894	0.774
LANN3_1		6	<b>0.363</b>	<b>0.409</b>	0.798	<b>0.778</b>
LANN4_1		7	0.354	0.938	0.804	0.277

LANN4_1	91-97Training: 98-00 Testing	7(NoOutliers)	<b>0.333</b>	0.748	<b>0.817</b>	<b>0.734</b>
L_JFM		8	0.172	0.196	0.910	0.926
LANN1_2	$P_t, P_{t-1}$ ,	8	0.296	0.694	0.839	0.796
LANN2_2	$E_t, E_{t-1}$ ,	7	<b>0.205</b>	<b>0.411</b>	<b>0.891</b>	<b>0.839</b>
LANN3_2	$E_{t-2}, E_{t-3}$	9	0.443	0.568	0.746	0.689
LANN4_2		9	0.490	1.242	0.715	0.426

In the above investigation, the training data set was from 1991 to 1998 and the testing data set was from 1999 to 2000. And the model L\_JFM is the data set from Jan., Feb. and Mar., which generate good results, and further confirms the assumption of homogenous data set could improve the performance of the ANN models.

For the first scheme, the performance of each local ANN models is superior to the global ANN model except LANN4\_1. That is perhaps due to some outliers in this data set. The fact of better results of LANNs further proves that clustering data set into more homogenous data set would improve the performance of ANN.

From the above table, the optimal network models for each cluster is identified and listed in Table 9.

*Table 9. Optimal Local ANN models*

	Optimal Input Patterns
GANN	$PE_t, PE_{t-1}, PE_{t-2}, PE_{t-3}, PE_{t-4}, PE_{t-5}$
LANN_1	$PE_t, PE_{t-1}, PE_{t-2}, PE_{t-3}, PE_{t-4}, PE_{t-5}$
LANN_2	$PE_t, PE_{t-1}, PE_{t-2}, PE_{t-3}, PE_{t-4}, PE_{t-5}$
LANN_3	$PE_t, PE_{t-1}, PE_{t-2}, PE_{t-3}, PE_{t-4}, PE_{t-5}$
LANN_4	$PE_t, PE_{t-1}, PE_{t-2}, PE_{t-3}, PE_{t-4}, PE_{t-5}$

In the summer period, when evaporation usually exceeds precipitation, the input parameters to LANN models should add more information about evaporation. While during wintertime, less evaporation happen and rainfall is the dominating factors, so the input vector should be formed mainly by past rainfall. The performance of LANN using data set of Jan., Feb., and Mar. improved further, which signals that data set from these three months are more similar than the previous classifications

Due to the generalisation property of ANN models, it can sometimes fail to simulate accurately all of flow stages such as baseflows, low flows, medium flows and high flows simultaneously. Therefore, to ensure acceptable results, it may be necessary to have separate networks, to forecast the different stage of the runoff, owing to the dynamic nature of the non-linear system. Then a local forecasting strategy should be presented, in which the term local is taken to mean a forecasting tool that is able to operate effectively at a particular range of runoff. In this strategy, runoff process (series) should be divided into several groups,

which are more homogenous than mixed all the data together. These groups have their own general characteristics.

The optimal neural network for each class is listed in Table 9. Compared with the results in Table 3, it can be seen that for each sub data set, the performance of LANN is better than that of the GANN. It can also be seen that there are different input patters for each LANN. (For dry period, evaporation is the dominant factor that influences the runoff, at least in our case according to the available data; for wet period, rainfall formed the main part of input parameters.)

The comparison of the results of best global ANN model and that of optimal local ANN models are illustrated in Figure 16.

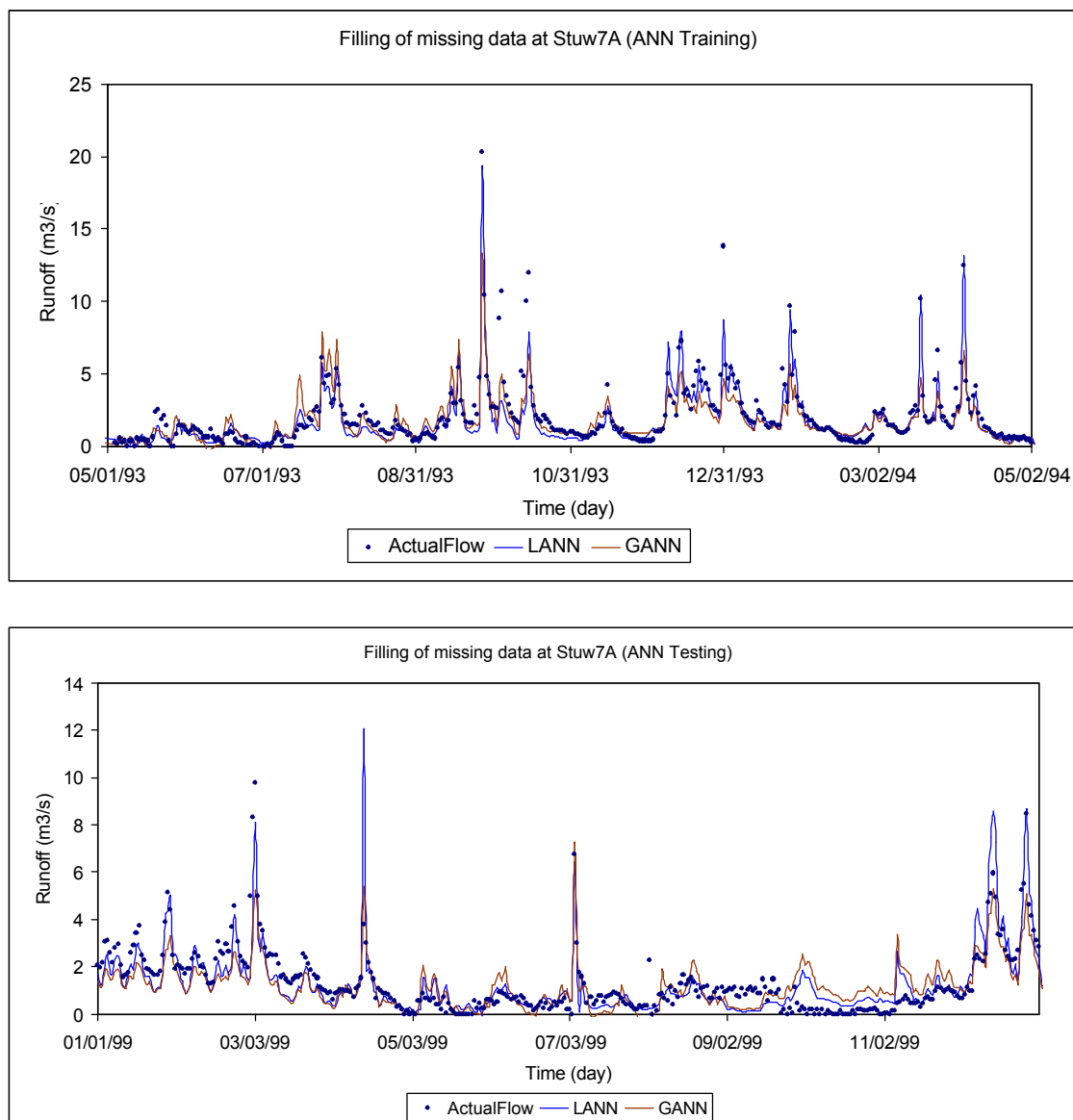


Figure 16 The hydrographs comparison of GANN and LANN

#### 5.4 Application of ANN for runoff forecasting at Rietberg

Continuous simulation and prediction of runoff plays an essential role in water resources management. It could offer information for optimal operation of control structures and provide effective support for decision making. There are many driving factors in the process of runoff generation and since different mechanisms dominate the different stages, namely base flow, low flow, median flow and high flow, the runoff generation process presents a higher degree of non - linearity. All those things together may make the work of forecasting runoff precisely not an easy matter.

In the previous section, ANN models are constructed to fill the missing runoff values using the continuous rainfall and evaporation series. But now, the past runoff sequences are also taken as part of input patterns and artificial neural network models are set up to perform runoff prediction task, that is, to predict the future discharge one time step ahead in order to carry out continuous forecasting work

The optimal number of hidden layers and the number of neurones in the hidden layers are normally obtained through experimentation with the available data set. Based on the results of the auto-correlation and cross-correlation analysis, daily discharge and rainfall values with lag-one to lag-five were considered as input to the ANN models. Eight different schemes were designed and experiments of trial and error extensively carried out to identify the optimal number of hidden nodes and other parameters.

For Rietberg, a continuous available daily discharge of 1997 was used as training data set, while the data between 1<sup>st</sup> of Dec., 1998 and 31<sup>st</sup> of Aug., 1999 was selected as testing data set to set up a prediction ANN model. The performance of each model is listed in Table 10.

Table 10 The performance of 10 schemes

Input patterns	Hidden nodes	NMSE		MSE		R	
		Training	Testing	Training	Testing	Training	Testing
Naïve Model		1.015	0.802	0.245	0.214	0.492	0.599
$Q_{t-1}$	5	0.624	0.621	0.151	0.166	0.613	0.616
$Q_{t-1}, Q_{t-2}$	5	0.462	<b>0.491</b>	0.588	<b>0.113</b>	0.734	<b>0.725</b>
$Q_{t-1}, Q_{t-2}, Q_{t-3}$	6	0.620	0.622	0.150	0.166	0.616	0.617
$PE_{ma3,t-1}, Q_{t-1}, Q_{t-2}$	6	0.321	0.561	0.078	0.146	0.826	0.664
$P_{t-1}, E_{t-1}, Q_{t-1}, Q_{t-2}$	4	0.275	0.766	0.066	0.200	0.852	0.538
$P_{t-5}, P_{t-4}, P_{t-3}, P_{t-2}, P_{t-1}, Q_{t-1}, Q_{t-2}, Q_{t-3}$	10	0.245	0.813	0.059	0.217	0.870	0.539
$P_{t-5}, P_{t-4}, P_{t-3}, P_{t-2}, P_{t-1}, Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}$	10	0.243	0.800	0.059	0.214	0.872	0.558
$P_{t-5}, P_{t-4}, P_{t-3}, P_{t-2}, P_{t-1}, P_t, Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}, Q_{t-5}$	13	0.181	<b>0.449</b>	0.044	0.120	0.905	<b>0.760</b>

The optimal neural network architecture is chosen based on the result in the above tables as the following:

$$Q_t = f(Q_{t-1}, Q_{t-2}) \quad (8)$$

Figure 17 shows plots of observed versus simulated discharge values of the optimal network output for both the training and testing period. From these results, one could see that the performance of the model could be considered satisfactory. Longer duration of training data and smaller time interval between the data could have resulted to even better performing networks.

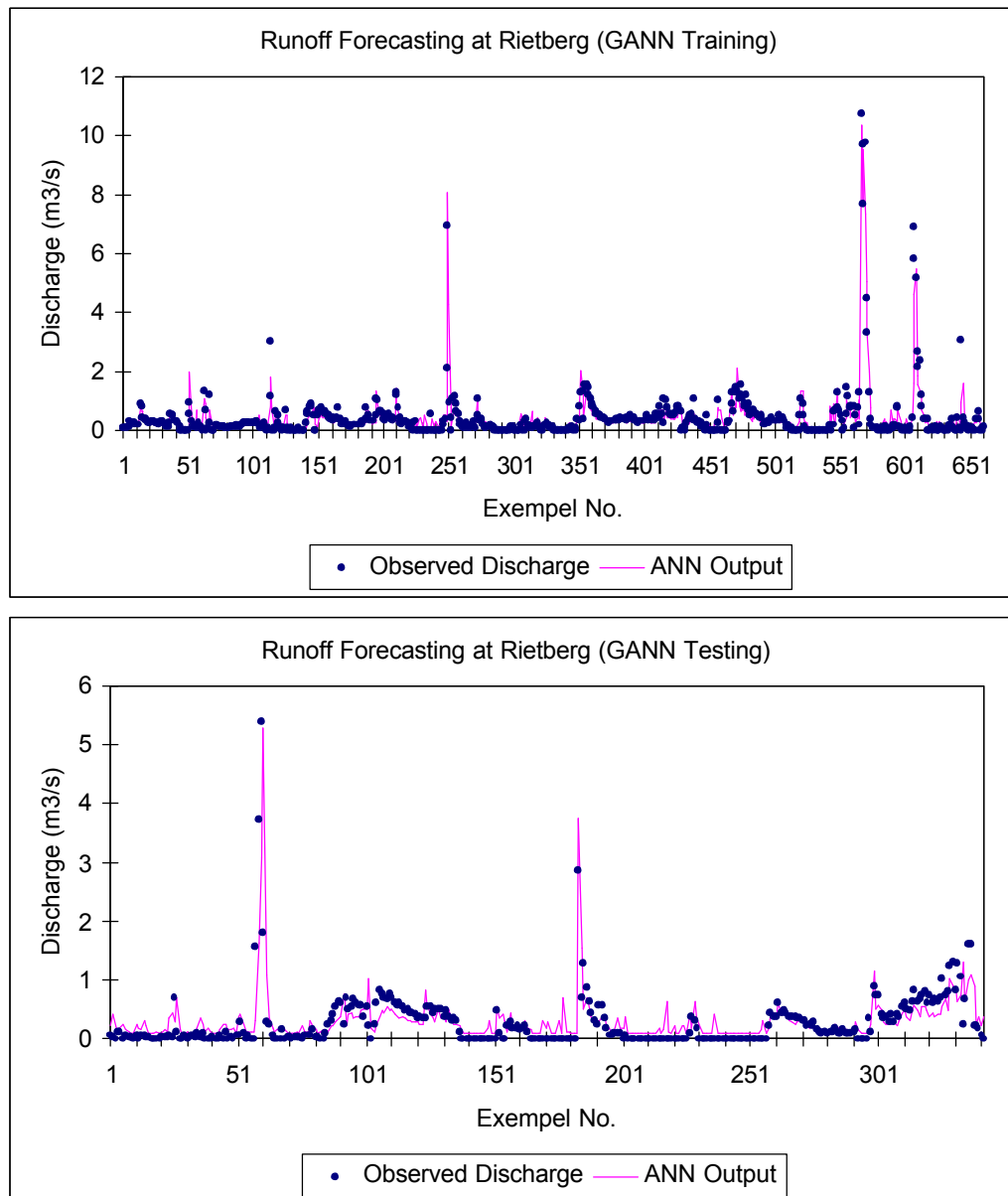


Figure 17 Prediction results of optimal MLPs model

### 5.5 Application of ANN to runoff forecasting at Stuw 7A

A continuous sequence daily discharge of Dec. 1998 to Feb. 2000 is used as training data set, while 1997 is selected as testing data set to set up an optimal ANN model for forecasting

runoff of Stuw7A. Different network architectures were considered and the performance of each model is listed in Table 11.

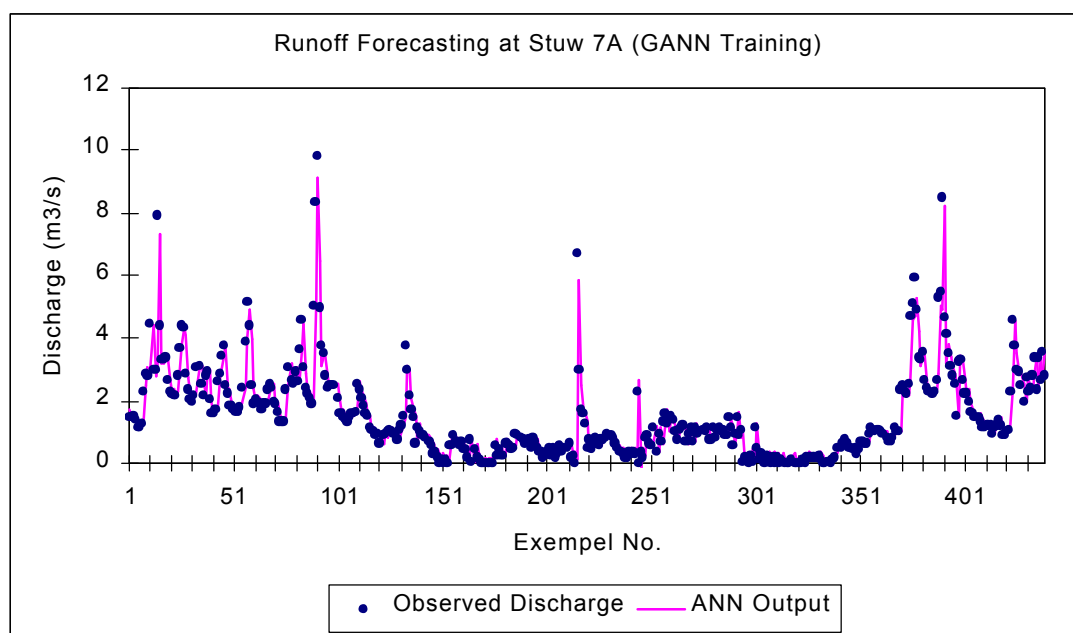
The best result is obtained when the input to the network consists of two previous discharges. Figure 18 shows the training and testing results of the optimal neural network whose structure could be expressed as following:

$$Q_t = f(Q_{t-1}, Q_{t-2}) \quad (9)$$

Table 11 The performance of 8 schemes

Input patterns	Hidden nodes	NMSE		MSE		r	
		Training	Testing	Training	Testing	Training	Testing
Naïve Model		0.319	0.243	0.624	0.195	0.840	0.879
$Q_{t-1}$	2	0.270	0.220	0.528	0.177	0.854	0.886
$Q_{t-1}, Q_{t-2}$	2	0.253	<b>0.212</b>	0.495	<b>0.171</b>	0.864	<b>0.890</b>
$Q_{t-1}, Q_{t-2}, Q_{t-3}$	7	0.228	0.246	0.449	0.198	0.878	0.873
$Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}$	8	0.215	0.286	0.420	0.230	0.886	0.855
$P_{t-4}, P_{t-3}, P_{t-2}, P_{t-1}, Q_{t-1}, Q_{t-2}, Q_{t-3}, Q_{t-4}$	9	0.168	0.321	0.328	0.259	0.912	0.839

This figure shows that performance of the trained ANN in forecasting discharge one-day ahead is quite well. The relatively better performance of ANN for runoff forecasting at Stuw7A could mainly be attributed to the more homogeneous and longer records of the training data.



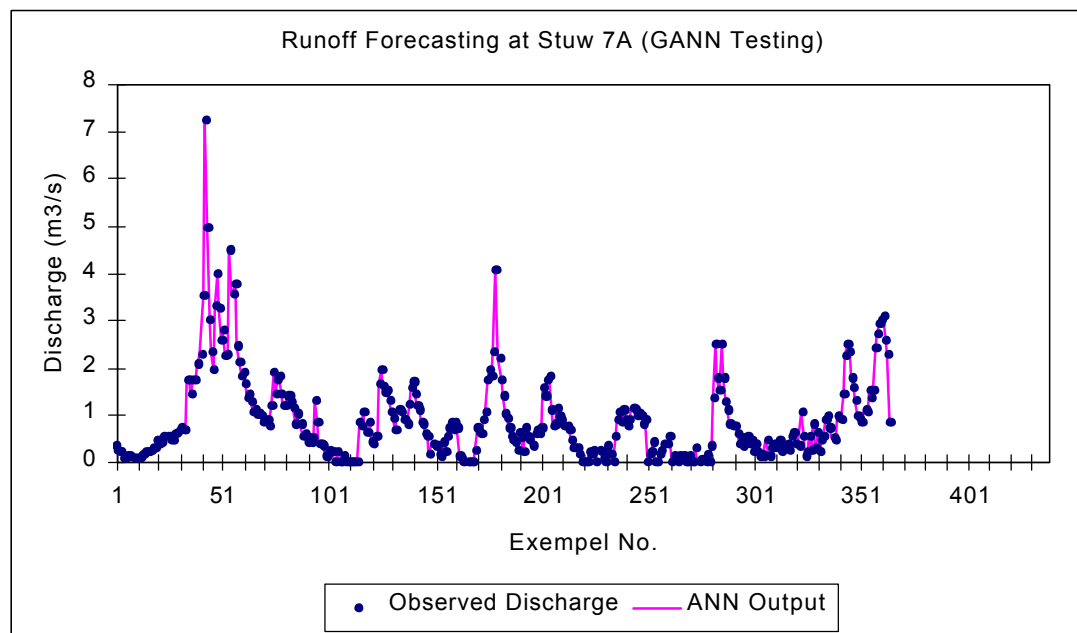


Figure 18 Results of the optimal ANN model

To see if any further improvement could be obtained, all the available data organised together to construct LANN models. Table 12 shows the performance of these ANN models.

Table 12. The performance of 2 LANN schemes

Models	Input Patterns	Hidden nodes	NMSE		r	
			Training	Testing	Training	Testing
LANN1_1	$Q_{t-1}, Q_{t-2}$	3	0.312	0.405	0.829	0.775
LANN2_1		3	0.419	<b>0.306</b>	0.762	<b>0.843</b>
LANN3_1		3	0.278	0.645	0.887	0.598
LANN4_1		4	0.346	<b>0.321</b>	0.810	<b>0.838</b>
LANN1_2	$P_{t-5}, P_{t-4}, P_{t-3},$	10	0.295	0.431	0.840	0.756
LANN2_2	$P_{t-2}, P_{t-1}, Q_{t-1},$	9	0.361	<b>0.217</b>	0.800	<b>0.887</b>
LANN3_2	$Q_{t-2}, Q_{t-3}$	10	0.299	0.670	0.864	0.578
LANN4_2		10	0.326	<b>0.279</b>	0.821	<b>0.851</b>

From the above table, one could find out that for two of the networks, one (LANN2) corresponding to the months of Mar\_Apr\_May and the other (LANN4) to the months of Sep\_Oct\_Nov, the performance has been improve when compared to than of GANNs'. However, the results of the other two LANNs did not show improvement.

In general, ANN models forecasts the magnitude of the baseflow quite well, but encounters some difficulty in forecasting the magnitude of the peak flows. That perhaps due to the rapid runoff compared to the model step size. It was also observed that for Stuw7A, the performance of the ANN models is superior to that of Rietberg.





## 6 Conclusion and Recommendations

In this research the applicability of artificial neural networks to fill missing discharge data and to forecast runoff was investigated. The area of application was the Salland region in the Netherlands, from which several time-series data on precipitation, evaporation, surface water level were available. Preliminary data analysis showed that the hydrological time series of all the three drainage areas considered has significant number of missing values and inconsistencies. Therefore, the final application was focussed only on two drainage areas (Rietberg and Stuw 7A) and time periods with reasonably consistent data.

Two methods of using ANN, namely global neural network (GANN) and local neural networks (LANN) were considered. GANN considers all available time series data in its entirety. In cases where relatively long time series data are available and smaller number of extreme events occur, this approach appeared not to be the best approach. The reason is that during the training process, rare information is considered as noise and is filtered out, while these rare events represent the important events for which we would like to obtain more accurate ANN model outputs. To be able to build LANN models, the complete time-series data has to be split into more homogeneous sub-sets so that the highly non-linear behaviour of the entire runoff process is captured in different classes for which the input-output relationships can be relatively simple.

In general, LANNs have outperformed the GANNs for both problems of filling missing data and runoff forecasting. Moreover, using short-term history of water system variables as inputs to the network gave the best results. Once the ANN models are built, they are used to estimate values for missing runoff data and forecast a one-day ahead discharge on the basis of available meteorological data such as measured rainfall, evaporation and discharge.

It must be mentioned that the operation of weirs and pumping stations in the area affects very much the homogeneity of input-output relationships in the data sets. As a result, ANN prediction of runoff values may not always match the monitored values. This could be due to the fact that manually operated weirs and discharge outlet structures control the flow in the drainage area and interfere with the natural flow and this, in its turn, affects the predictability of system behavior. However, the success of these ANN models in replicating the systems behaviour could be further improved by including information about the operational data of those regulating structures. The results could also be improved by classifying the data, not only by seasonal variations, but also by the magnitude of runoff events in the database. Moreover, it is important to frequently update the models by additional training or complete retraining every time new data set is available so that the models reflect the latest state of the system being modelled.

In general, the case studies on the catchments in Salland clearly demonstrated the applicability of artificial neural networks for runoff forecasting and filling of missing data in hydrological time series based on meteorological and other hydrological data. Recommendations are also given on how to further improve the result by including additional relevant information during the model identification processes.



## 7 Reference:

- ASCE Task Committee on ANNs in Hydrology, 2000b. ANNs in Hydrology. II. Hydrologic Applicaitons, *Journal of Hydrologic Engineering*, Vol. 5, No. 2.
- Brater, E. F., 1976, *Handbook of hydraulics for the solution of hydraulic engineering problems*, New York, McGraw-Hill.
- Dibike Y.B., and Solomatine D. (2001) River Flow Forecasting Using Artificial Neural Networks, *Journal of Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere*, Vol. 26, No.1, pp. 1-8.
- Hannah D. M., Smith B. P. G., Gurnell A. M. and McGregor G. R., 2000. An approach to hydrograph classification, *Hydrological Processes* 14, 317 – 338
- Khalil M., Panu U. S., Lennox W.C., 2001. Groups and neural networks based streamflow data infilling procedures, *Journal of Hydrology* 241, 153 – 176
- Lobbrecht, A.H., Solomatine, D.P., B. van der Wal. Kunstmatige intelligentie in het waterbeheer (Artificial intelligence in water management). *Het Waterschap*, No. 14, July 2002, pp. 644-649 (in Dutch).
- Lobbrecht, A.H., Solomatine, D.P. Machine learning in real-time control of water systems. *Urban Water* 4, 2002, 283-289.
- Zealand C. M., Burn D. H., Simonovic S. P., 1999. Short term streamflow forecasting using artificial neural networks, *Journal of Hydrology* 214
- Solomatine, D.P. Applications of data-driven modelling and machine learning in control of water resources. In: *Computational intelligence in control*, M. Mohammadian, R.A. Sarker and X. Yao (eds.). Idea Group Publishing, 2002, pp. 197 – 217.
- Zhang B, Govindaraju RS. 2000. Modular neural networks for watershed runoff. In *Artificial Neural Networks in Hydrology*, Govindaraju RS, Rao A (eds). Kluwer Academic Publishers: Dordrecht, The Netherlands; 73–91.



**Artificial Neural Networks and Fuzzy Logic  
Systems for Model Based Control:  
Application to the Water System of Overwaard**



**Project report  
Part III**

By  
A.H. Lobbrecht  
Y.B. Dibike  
D.P. Solomatine

**Delft  
September, 2002**



## Contents of Part 3

<b>1 INTRODUCTION .....</b>	<b>133</b>
1.1 GENERAL .....	133
1.2 THE STUDY AREA .....	133
1.3 OBJECTIVE OF THE STUDY .....	135
<b>2 SIMULATION AND CONTROL OF WATER SYSTEMS .....</b>	<b>137</b>
2.1 SIMULATION OF WATER SYSTEMS .....	137
2.2 WATER SYSTEM CONTROL .....	140
<b>3 DATA ANALYSIS .....</b>	<b>143</b>
3.1 HYDROLOGICAL DATA .....	143
3.2 BOUNDARY DATA .....	146
3.3 DATA ON STATE AND CONTROL VARIABLES .....	147
<b>4 AQUARIUS MODEL FOR OVERWAARD .....</b>	<b>149</b>
4.1 AQUARIUS MODELLING SYSTEM FOR SIMULATION AND CONTROL .....	149
4.2 SETTING UP OF AQUARIUS MODEL OF OVERWAARD .....	150
4.3 CALIBRATION OF AQUARIUS MODEL OF OVERWAARD .....	153
4.4 CENTRAL DYNAMIC CONTROL WITH AQUARIUS .....	155
<b>5 ARTIFICIAL NEURAL NETWORKS AND FUZZY ADAPTIVE SYSTEMS .....</b>	<b>159</b>
5.1 ARTIFICIAL NEURAL NETWORKS .....	159
<i>Multi-layer Perceptron Network (MLP)</i> .....	160
<i>Learning in MLP networks (training)</i> .....	161
5.2 FUZZY LOGIC AND FUZZY ADAPTIVE SYSTEMS .....	162
<b>6 APPLICATION OF ANN AND FAS FOR OPTIMAL CONTROL .....</b>	<b>165</b>
6.1 MODEL-BASED CONTROL .....	165
6.2 TRAINING OF ANN AND FAS .....	166
6.3 DEVELOPING EXTERNAL CONTROLLER .....	168
6.4 RESULTS AND DISCUSSION .....	169
<b>7 CONCLUSIONS AND RECOMMENDATIONS .....</b>	<b>171</b>
<b>8. REFERENCES .....</b>	<b>173</b>





# 1 Introduction

## 1.1 General

Modern water management is characterised by considering water systems in their entirety together with all influencing factors and other related systems. As a result, current policy objectives for water management focus on the creation and maintenance of a sustainable living environment, taking into account all demands made on the water system by the different interest. In general, operation and maintenance of regional water resources system concern the optimal resources allocation for various interest groups in the system at the same time. Hence, to get an accurate overall picture of a water system state, which can be used for optimal operation and maintenance, it is necessary to take the conflicting criteria into account. Therefore, the problems of management of water resource systems are usually posed as multi-criterial problems which has to be solved (numerically) to arrive at the optimal condition. In order to address such problems, a decision support system (DSS) is usually built that is capable of generating several control strategies aimed at optimal control at local and centralised level. However, one of the problems in using such DSS is the high computational time needed to generate an optimal control strategy, and its sensitivity to some of the parameters and its variables.

Recent development in the field of artificial intelligence (AI) techniques are helping to solve various problems of water resources modelling and management. These techniques have shown their potential as an alternative approach to conventional controllers. Especially artificial neural networks (ANN) and fuzzy adaptive systems (FAS) appear to be efficient alternatives to using optimal control algorithms in real-time tasks (a comprehensive literature review on applications of ANN and FAS techniques to problems in water management are presented in the first project report). The relation between the optimal decision or action and the influencing parameters can be learned by neural networks and fuzzy adaptive systems. Once identified, then it is possible to use these relations for deriving the decision and control actions in real-time. They can also be combined with the conventional controllers to enable better handling of complex real-life problems.

## 1.2 The study area

This study is done for an area called Overwaard, a drainage basin located in South-Holland, the Netherlands and managed by the water board of “Hoogheemraadschap van de Alblasserwaard en de Vijfheerenlanden” (see Fig.1). The Water Board has three water management areas in the region with a total of over 38,000 hectares and 210,000 inhabitants spread over 13 city councils. Out of this, the Overwaard drainage basin constitutes approximately 15,000 hectares. The area is, on average, two meters below sea level and it has been protected by elaborate system of dikes since the fourteenth century. The use of windmills to pump water out of the low lying polders dates back to 1740 and some can still be seen in the area. However, at the present time they are all replaced by a system of pumping stations operated by electrical power.

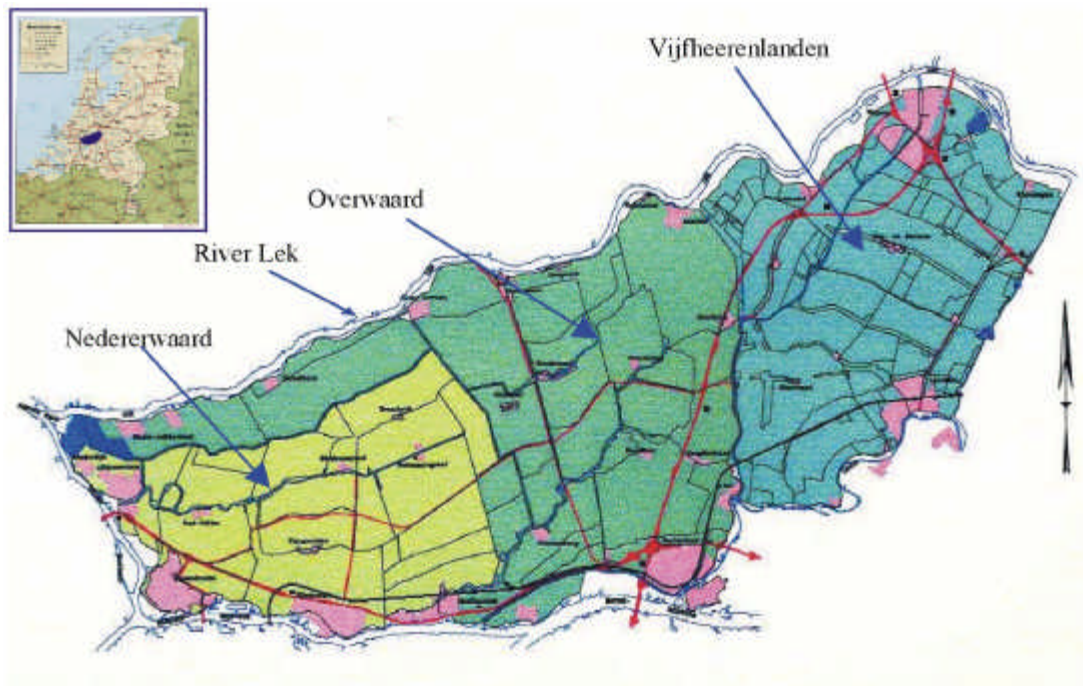


Figure 1. Area managed by the water board of Alblasserwaard en de Vijfheerenlanden

In general, the water system of Overwaard (Fig. 2) consists of the following items:

- An upper basin ('hoge Boezem', ~98 ha)
- A lower basin ('lage Boezem', ~170ha and 58 km long)
- Main water ways ('Hoofdwatgangen' of 280 km long)
- 22 drainage areas ('bemaalingsgebieden', total area of ~15,000 ha)
- 21 polder pumping stations ('gemalen', total capacity ~ 24 m<sup>3</sup>/sec)
- one main pumping station (total capacity 25 m<sup>3</sup>/sec)
- Sluice gate ('uitwateringssluis')

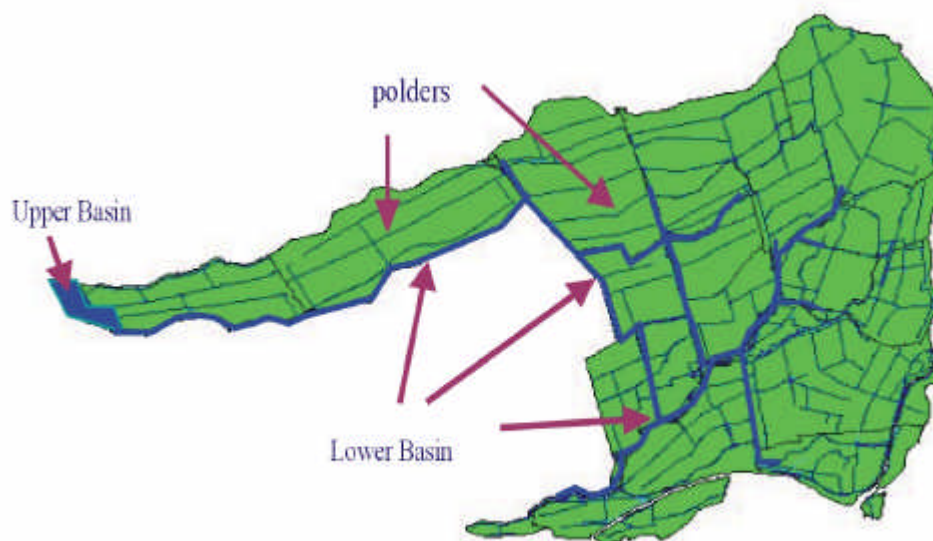


Figure 2. Drainage area of Overwaard along with the upper and lower storage basins

The soil type in the Overwaard area consists mainly of peat and marine clay. The land use is mainly for agriculture and especially livestock farming. During excess precipitation the surface water levels in the polders start to rise. When the levels reach a certain (pre-defined) level, then each polder pumping station will start pumping water in to the lower basin storage area. Similarly, when the water level in the lower basin reaches at a certain pre-specified point, then the main pumping station will start pumping water to the upper basin storage area, which will in its turn discharge the excess water through the sluice gates to the River Lek during low tide. During summer time, the water levels in the polder surface water system may reach lower limits. In such cases, the flow direction will be reversed and water is supplied from the river into each polder area through inlets.

### **1.3 Objective of the study**

There two main objectives for this study and they can be described briefly as follows:

- (i) To build a conceptual simulation model of the water system at Overwaard, which can be used for operational management purposes.
- (ii) To investigate the possibilities of using artificial intelligent techniques such as ANN and Fuzzy logic systems to improve the operational performance of the model for better control of the water system.



## 2 Simulation and Control of Water Systems

Water systems consist of hydrological units, which interact via natural and artificial water flow paths. Several aspects of water systems have to be considered to solve water-system control problems. For instance, the introduction of control elements in a water system changes the original hydrological system and split it up into separate areas, each consisting of one or more subsystems (see Fig. 3). The major issue is therefore the allocation of scarce or plentiful water resources to the various interests at the right time while satisfying the requirements of the interests as much as possible.

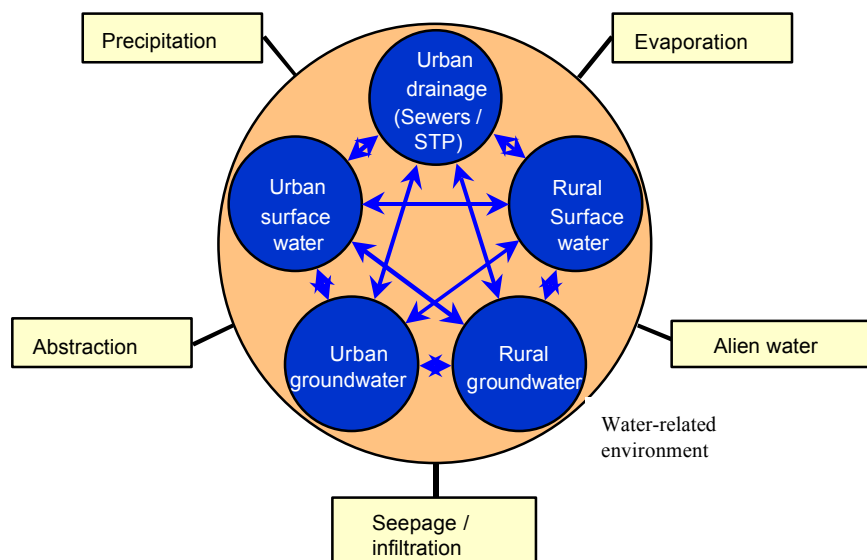


Figure 3. Different hydrological units and external loads in a water system

### 2.1 Simulation of Water Systems

A water system can be described by subsystems and flow elements whereby subsystems interact via flow elements. Flow elements can be categorised into controllable flow elements (or regulating structures) such as pumping stations and controllable inlets; fixed flow elements (or fixed structures) such as fixed weir; and free flow elements such as canals and regional groundwater flow. The number of subsystems and flow elements that have to be considered depends on the amount of detail needed to describe the corresponding processes. Moreover, the hydrology of a particular water system generally cannot be separated completely from surrounding water system. Therefore, interaction with neighbouring systems are included in the water-system description itself by formulating the boundary conditions that influence or are influenced by the water system at the water-system boundary. Figure 4 shows a typical regional water system, including various elements mentioned above.

In a regional water system, various types of surfaces can be found each having associated runoff processes that are of importance to regional water-system control. For instance, the subsystems of an urban area can incorporate rapid and slow runoff processes, such as rapid runoff via sewer systems and slow runoff via the permeable subsurface of green belts. In general, the main surface types that affect runoff can be distinguished as pervious, semi-impervious or impervious as shown in Figure 5. Here, such runoff processes are classified on

the basis of their specific characteristics and not on the basis of their geographical location in the water system. The interaction between the several subsystems are represented by various equations such as mass balance equations, equations for saturated or unsaturated flow (Darcy or Richards equation) or equations for sewer and surface water flows (For detailed mathematical descriptions refer to Lobrecht, 1997)

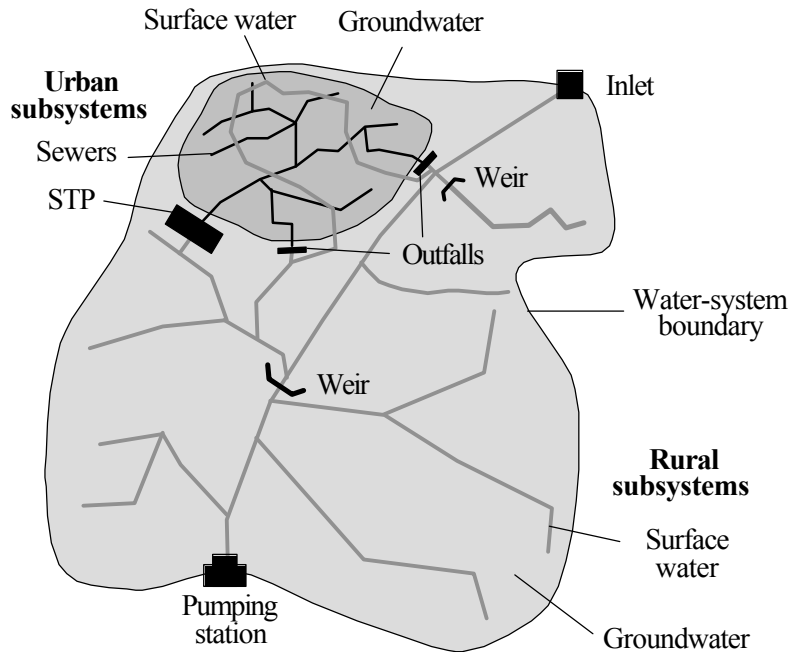


Figure 4 Schematic view of flow elements in a typical water system.

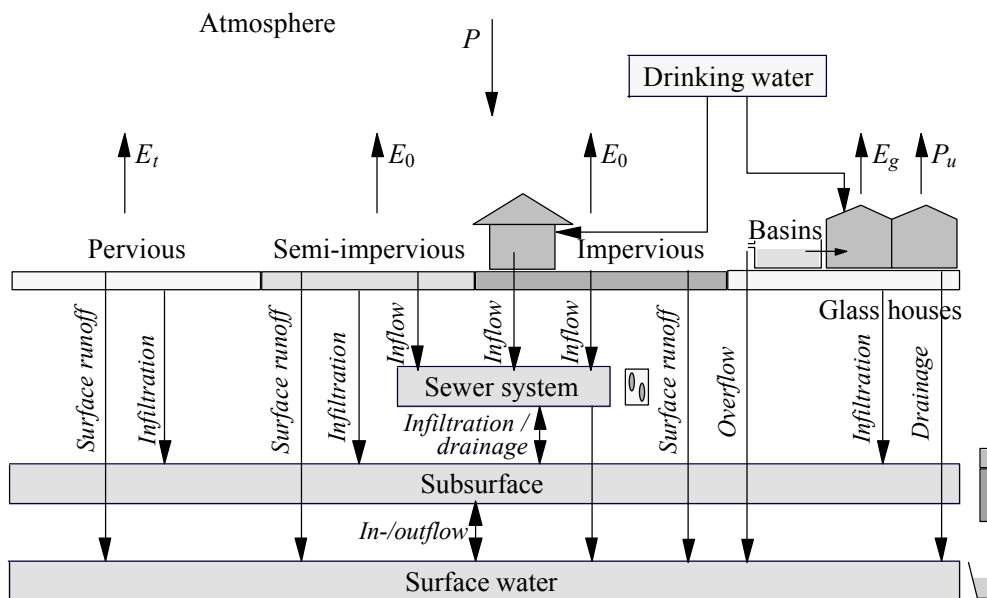


Figure 5. Runoff from various surface types

In addition to the natural interaction between subsystems described above, water systems also interact through various flow elements. Pumping stations are the main regulating structures in polder areas and large pumping stations usually have several pumping units and pumping stations can be either manually or automatically controlled. In general, a locally controlled

pumping station operates on the basis of only the upstream or the downstream water level (see Fig. 6). In practice, restrictions are incorporated in the control of pumping stations. Pumping stations that drain polders and discharge to storage basins generally have extra conditions governing their operation. One restriction is the upstream water level in the storage basin. If this water level has reached an upper limit and is still rising, the water manager of the storage basin can impose a milling stop. As a consequence, pumping stations should be switched off. In case the pumps of a pumping station are driven by electrical energy, special measures may be taken to minimize energy costs. In general, the night tariff for electricity is lower than the day tariff. In that case, automated pumps should preferably operate during the night. To accomplish this, lower switching-on and -off levels can be set for the night. Some electricity companies apply additional high tariffs for energy use during peak hours. In general, pumps are switched off during these hours. Under extreme conditions, the water manager may still decide to use these hours for pumping.

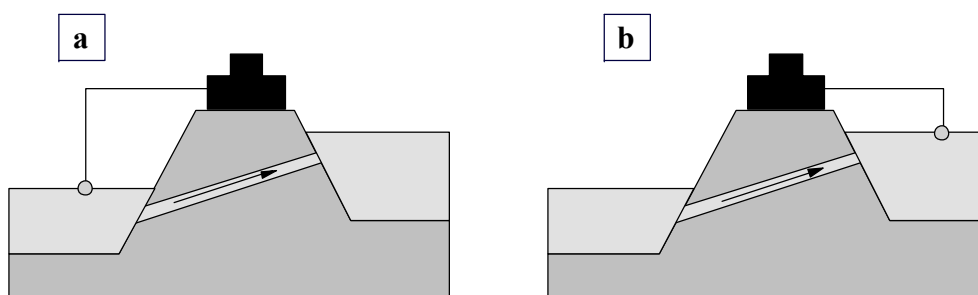


Figure 6. Upstream-controlled (a) and downstream-controlled (b) pumping station

Other flow elements that should also be considered during simulation of a water system are weirs, sluices, inlets and outlets. Weirs are used in water management in both polder and hilly areas, but they are mainly found in hilly areas. A weir can either have a fixed crest level or an automatically controlled crest level. In the latter case, the water level upstream or downstream is controlled by means of a mechanical or electrical unit. In general, electrical controllers are used these days for weir regulation and this requires on-line water-level measuring. The measured signal is fed to the controller, which determines the control action of the weir. The control action is electronically sent to the driving device of the weir. This device can adjust the weir in an upward or downward direction.

Sluice structure consists of one or more culverts, each of which can be closed off by a sliding gate. Most spill sluices along the coast and tidal rivers in the Netherlands are of the sliding-gate type. The discharge of a sluice can be described by various water-level situations on both the upstream and downstream sides. A controlled sluice can be opened as soon as the upstream surface-water level is higher than the downstream water level. An automatically operated sluice requires water-level measuring on both sides of the structure. In general, spill sluices have several gates which can be operated separately and be fixed at various heights. This indirectly allows an operator or computer to determine the discharge.

Inlet and outlet structures function very similar to sluices. Inlets and outlets are used for water-quantity and water-quality control. The function of inlets and outlets is very similar and they can be described together using the term 'inlets' only. Inlets are generally culvert-type structures that can control the downstream or upstream surface-water level. Inlets discharge by gravity. If the flow via the inlet depends on the water levels upstream and possibly also downstream, the sluice flow element should be used.

## 2.2 Water System Control

Designing and operating water systems is a matter of capacity allocation. The factual reason for water level rise and the subsequent flooding or overflow could be a lack of enough storage or discharge capacity in a subsystem such as surface-water, groundwater and sewer systems. The occurrence of such undesirable situations could be prevented by larger storage or discharge capacities and this will intern reduce corresponding frequency of system failure. However, such a reduction in the frequency of system failure can also be achieved by other means. Research shows that usually not all the available capacity of a water system is used at the moment of failure and that unused capacity usually remain in the system (Schilling, 1991). All subsystems of a water system rarely fail at the same moment. Therefore, the temporarily unused capacity of one subsystem can be used in favour of another subsystem by making use of the available regulating structures that is operated regularly, either manually or automatically, adjusting the flow in the system.

Automation of routine tasks is one of the first steps towards improving water system control. However, automation alone is not sufficient to meet many of the requirements. The awareness is growing that a weighed form of water-system control, in which automation plays a role, is necessary. In this respect, three evolutionary steps can be distinguished that will eventually lead to weighed control of a water system (see Fig. 7):

1. local control,
2. central control,
3. dynamic control

Local control involves a single regulating structure in a water system and is executed on the basis of monitoring data gathered in the vicinity of that structure. Local control take place on the basis of the standards that has been set for each subsystem. This form of control is practised in many water systems by pumping stations that control surface-water level and weirs that control upstream water levels.

Central control involve one or more regulating structures and is executed on the basis of data from more than one location in the water system. Several subsystems can be involved in central control. Similar to local control, central control take place on the basis of pre-set standards. However, since a better picture is available of the water-system, the required water-system state can be determined, avoiding unnecessary or contradictory local control actions. Such central control mechanism generally implies logical control rules.

Dynamic control is a specific mode of central control, in which control actions are based on the time-varying requirements of interests in a water system, the water-system load and the dynamic process in the water system. For instance, a safety interest may require flood prevention, a recreational interest good water quality, industry sufficient cooling water, etc. Dynamic control incorporates a mechanism that enables continuous weighing of such interests present in the various subsystems. Meeting interest requirements can often be expressed in terms of physical water-system variables such as surface and ground water levels and water quality requirements. In dynamic control, deterministic optimisation technique is used to determine the best control strategy that can meet the requirements of all interests as well as possible. The formulation of the objective of control, the water system relationships and the limits to water-system variables together form the optimisation problem. The control problem for the general water system can be expressed by the following equations (Lobbrecht, 1997):



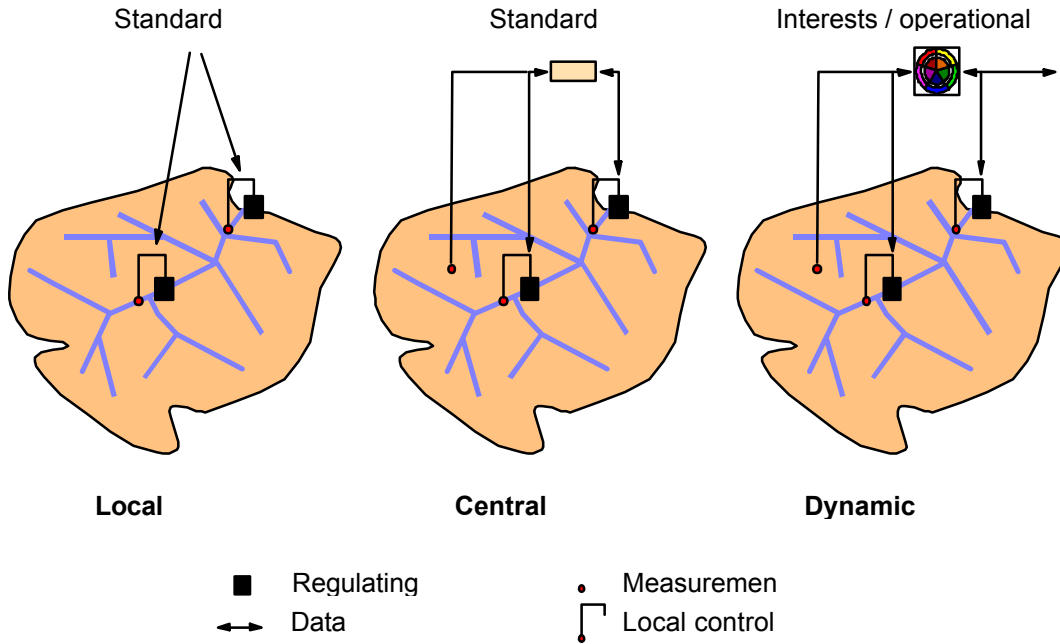


Figure 7 Three different types of water system control

$$\begin{aligned}
 &\text{Minimize} && Z(\bar{x}, \bar{u}), \\
 &\text{Subject to} && g_i(\bar{x}, \bar{u}) \leq 0, \quad i \in 1, \dots, l; \\
 & && x_{i,j} \leq x_j \leq x_{u,j}, \quad j \in 1, \dots, m; \\
 & && u_{i,k} \leq u_k \leq u_{u,k}, \quad k \in 1, \dots, n;
 \end{aligned} \tag{1}$$

in which

- $Z(\bar{x}, \bar{u})$  : objective function;
- $g_i(\bar{x}, \bar{u})$  : constrains;
- $\bar{x}$  : vector of state variables;
- $\bar{u}$  : vector of control variables;
- $x_{i,j}, x_{u,j}$  : lower and upper limits on state variable  $x_j$
- $u_{i,k}, u_{u,k}$  : lower and upper limits on control variable  $u_j$

The objective of water-system control is to satisfy the requirements of all interests present in that water system. The satisfaction of these interests can be expressed in water system variables such as surface and ground water levels. The requirement for each interest can be incorporated in the objective function  $Z$  by damage functions  $D$  of system state variables  $x$ . Each damage function penalises the deviation of the state which is required by an interest in a subsystem. The objective function for one time step  $\Delta t$  of the control horizon can then be written as:

$$Z(\bar{x}, \bar{u}) = \sum_{i=1}^m W_{ai} \sum_{j=1}^n R_{di,j} D_{i,j}(x) + \sum_{k=1}^r W_{rk} u_k \tag{2}$$

where  $D_{i,j}(x)$  represents damage function  $j$  for water subsystem  $i$ ; coefficient  $W_{ai}$  represents the weight of an area in the objective function and coefficient  $R_{di,j}$  expresses the relative importance of interest  $j$  in water subsystem  $i$  within a particular area. Additional penalties for

operation of regulating structures are described by  $W_{rk}$  and can be used to prevent unnecessary operation of structural units. Damage can be assigned to important variables such as surface and ground water levels (see Fig. 8). One or more damage functions can be used to represent an interest and each damage function incorporates a penalty  $p$  or a negative gain to the overall objective of control. The penalty is zero if a subsystem is in its desired state and increases with deviation from the desired state.

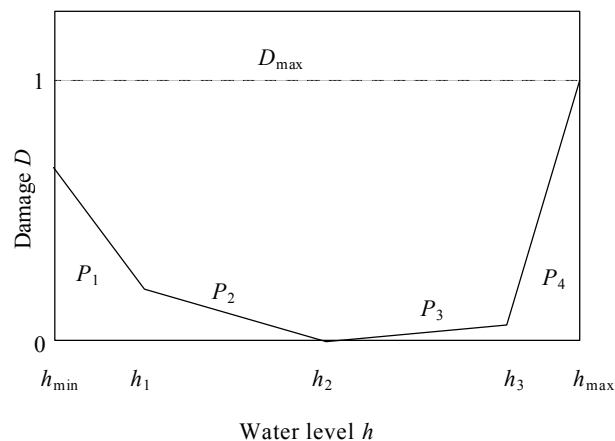


Figure 8. A typical damage function

The operational control problem in this context is solved by determining the best control actions for the moment and for some time ahead, on the basis of given historical, present and predicted system load. In this way, dynamic control allows an optimal use of the available system capacity both under normal and exceptional conditions. Using dynamic control during extreme hydrological conditions helps avoid failure in each subsystem as much as possible by optimal use of the available water-system capacity. The weighing mechanism used ensures that if failure cannot be prevented, the least important interests fail first and the most important once last. The dynamic-control method is incorporated in the software package AQUARIUS, which has been developed at TU Delft and used in this study.

### 3 Data Analysis

The first task of any modelling activity for the management of water resources is to collect as many historical (measured) data regarding the study area as possible. This may include meteorological data such as rainfall and evaporation; data regarding the drainage basins such as soil types, surface area of land and water bodies, ground water levels; boundary data such as upstream and downstream water levels of adjoining water systems; the various operational data for such control structures as pumping stations, sluice gates and inlet structures, etc. This types of data are necessary to build and calibrate the model in such a way that it would represent the study area as much as possible. Table 1 shows the various data collected and used to build and calibrate the water resources system of Overwaard. Other sources such as land use maps, soil maps, map of the main waterways, maps of hydraulic and other structures and maps of external flows were also used to obtain more quantitative and qualitative information.

*Table 1 Summary of the available historical data for the study area.*

Data type	Interval	Duration	Location
Rainfall	15 minutes	1/1/1995 till 7/6/2001	Land v.d. Zes Molens
Rainfall	1 day	1/1/1995 till 12/31/2000	Average of the three nearby stations
Reference Evapo-transpiration	1 day	1/1/1995 till 5/31/2001	Her wijnen
Water level of River Lek	15 minutes	3/1/1996 till 7/6/2001	River Lek d/s of sluice gate
Surface water level of the upper basins	15 minutes	1/1/1996 till 12/31/1998	Upper basin near the main pumping station
Surface water level of the lower basins	15 minutes	1/1/1996 till 12/31/1998	u/s of each polder pumping station
pump status at the main pumping station	15 minutes	1/1/1996 till 12/31/1998	at the main pumping station
Surface water levels and pumping status at each of the drainage areas	15 minutes	1/1/1995 till 12/31/1998	at each of the drainage areas
Groundwater levels at two places	Monthly	May 1995 till October 2000	in Land v. d. Zes Molens and Nieuw Goudriaans
Water supply through the main inlet	Daily	May 1995 till December 2000	Near the main pumping station

#### 3.1 Hydrological Data

Precipitation and evaporation are the main hydrological loads that would act upon a water system and are the main driving forces in the simulation of most water systems. For the simulation study at Overwaard, precipitation data from the stations at Land v.d. Zes Molens, Groot Ammers, Oud Alblas and Gorinchem and daily reference evaporation data from the station at Her wijnen covering the period from 1995 till 2000 were available. The time series

of precipitation data from the station at Land v.d. Zes Molens are at 15 minutes interval while the precipitation data from the other three stations are of daily interval. During this period, the annual rainfall generally varies between 724 mm and 1125 mm with yearly average of around 844 mm while the annual reference evaporation varies between 526 mm and 613 mm with yearly average of around 574 mm (see Fig. 9).

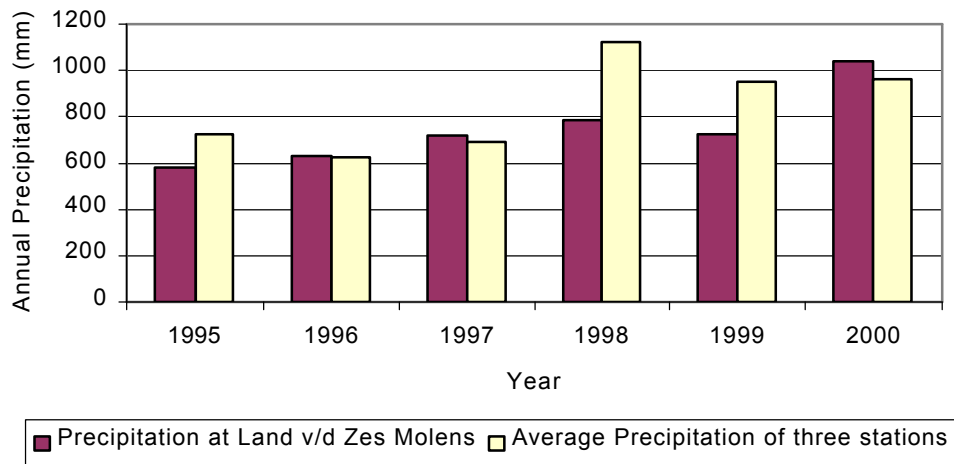
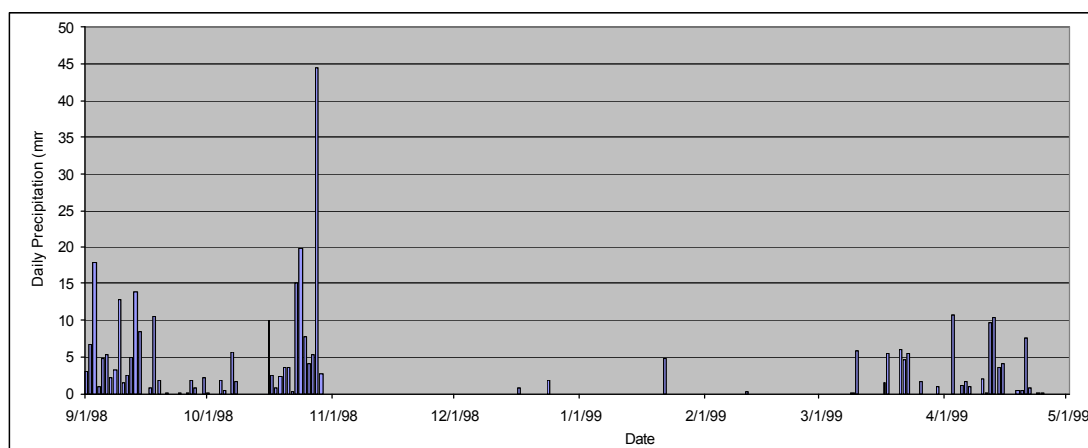


Figure 9. Annual precipitation at Land v/d Zes Molens and the other three nearby stations

In general, the 15-minute precipitation data at Land v.d. Zes Molens seems to be more appropriate for model simulation. However, the plot on Figure 9 reveals that, when compared to the average of the precipitation data from the other three nearby stations, the data from Land v.d. Zes Molens underestimate the total precipitation especially for the years 1998 and 1999. A closer look on Figure 10a and 10b also shows that the discrepancy occurs mainly during the period between September 1998 and April 1999, a period during which relatively heavier intensity of precipitation occurred. To correct for this discrepancy, the 15-minute precipitation data at Land v.d. Zes Molens was first reconstructed by proportionally scaling its values based on the daily average of the precipitation data from the three other nearby stations. Even though it may not be perfect, this approach will ensure the total volume of flow to be more consistent while allowing us to use the time series data with 15-minutes interval for simulation experiments. The daily reference evaporation data at Herwijnen was more or less consistent except the few segments of missing data which were filled by linear interpolation.



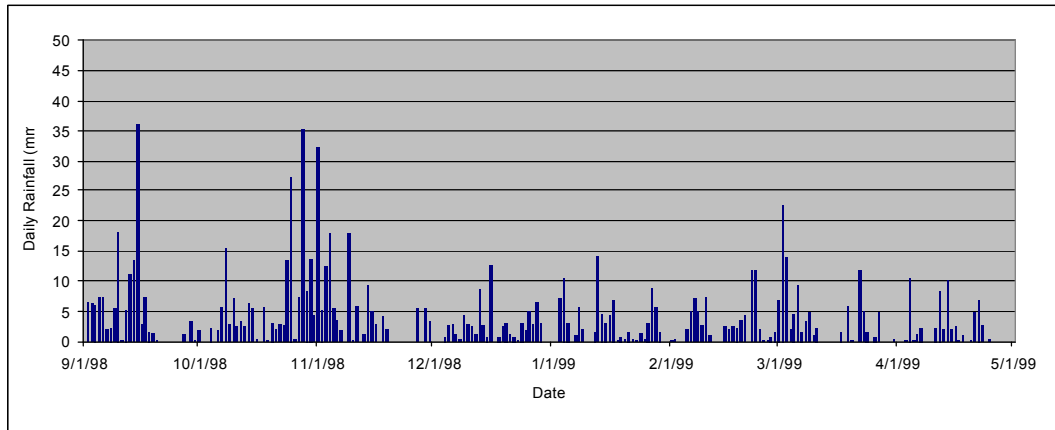


Figure 10 (a,b). Comparison of daily precipitation records at (a) Land v.d Zes Molens and (b) the average of the three nearby stations.

The daily precipitation and reference evaporation data are shown on Figure 11 and 12 while the average monthly data are plotted in Figure 13. The plots show that maximum and minimum average monthly precipitation occurred in June and April respectively while the maximum and minimum average monthly evaporation occurred in June and December respectively.

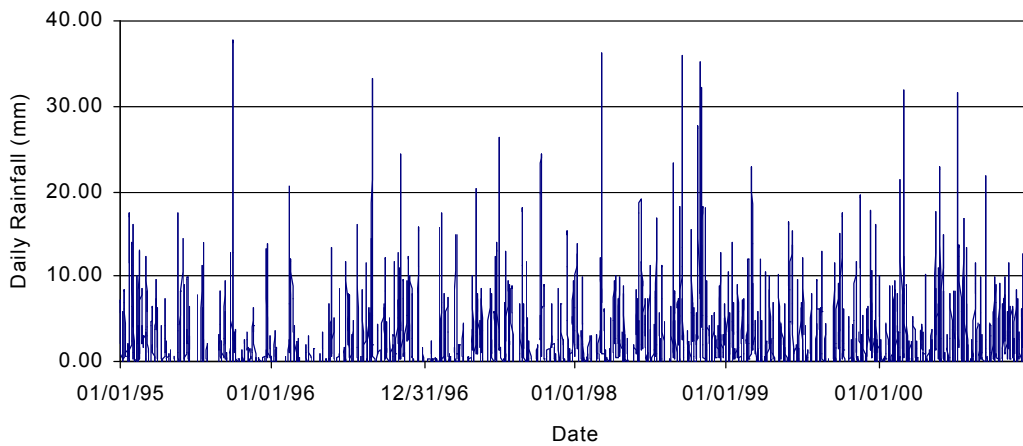


Figure 11. Daily precipitation data

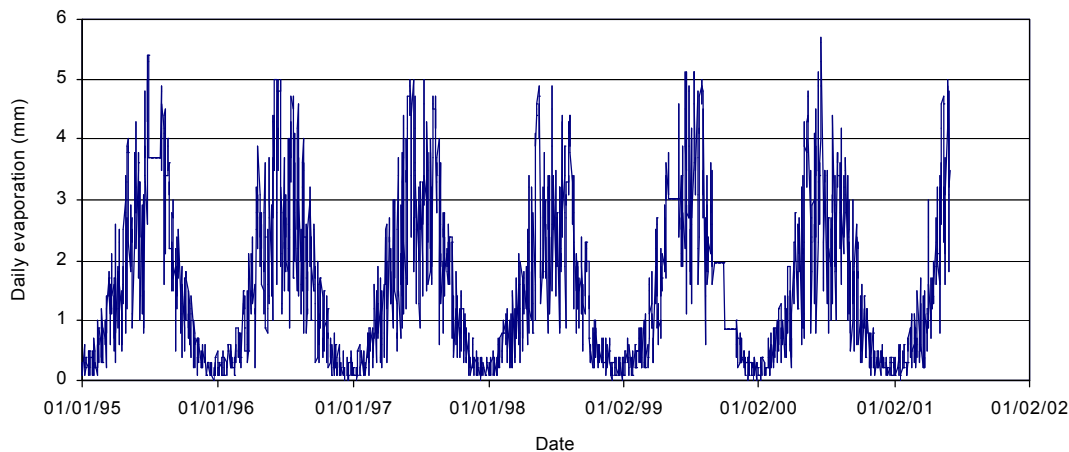


Figure 12. Daily reference evapotranspiration data

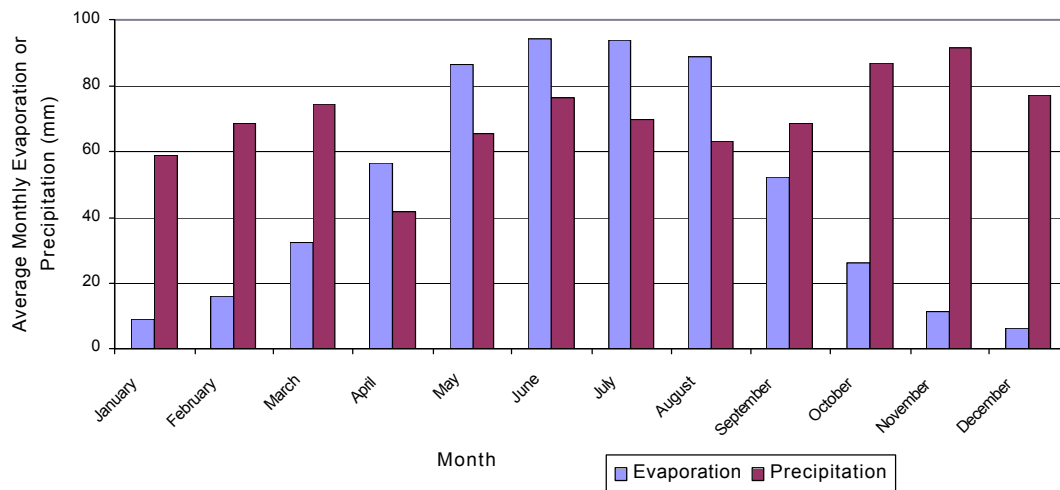


Figure 13. Average monthly evaporation and precipitation data

### 3.2 Boundary Data

It has been mentioned earlier that the hydrology of a particular water system cannot be separated completely from the surrounding water system and such interactions are included in the water system description itself, by formulating the boundary conditions that influence the water system at the water system boundary. The main external influence on the water system of Overwaard and hence the main boundary condition is the water level in the River Lek which is, in its turn, influenced strongly by the tidal cycle in the North Sea. The range of water level in the river near the sluice gate is between N.A.P.  $-1\text{m}$  and N.A.P.  $+2.5\text{m}$  with mean water level of around N.A.P.  $+0.5\text{m}$  and it has an average tidal period of around 12.5 hours. A segment of this boundary data during relatively high storm season is shown in Figure 14.

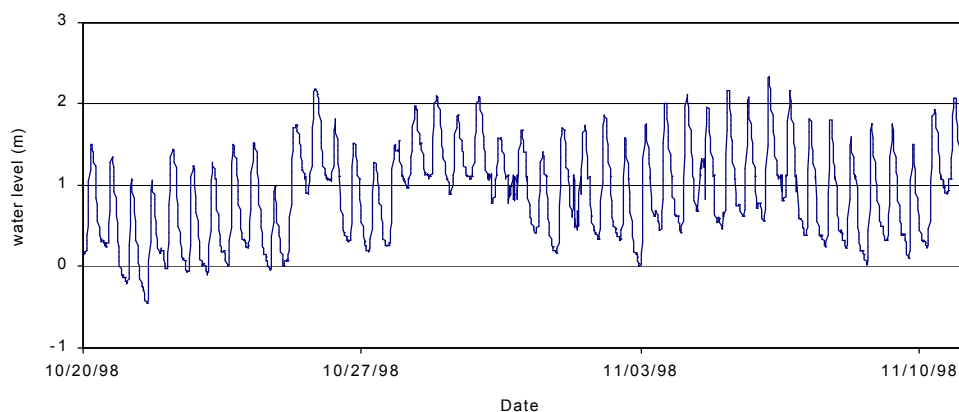


Figure 14. Water level of the River Lek near the sluice gate of Overwaard

Excess water from the upper basin is discharged to the river Lek through the sluice gates only during low tide periods when the water level in the river is lower than that of the basin. During winter, the sluice gets remain closed whenever the water level in the river is higher than that of the upper basin so as to prevent water from flowing into the drainage basin while during summer time water is supplied from the river into the basin through the inlet structure.

### 3.3 Data on state and control variables

In addition to the precipitation, evapotranspiration and boundary data, observed data on state and control variables were also collected. These are time series data of surface water level (in both the upper and lower basins) and pumping status (hence discharge) of the main pumping station between the two basins, all of which observed at 15 minutes interval. Similarly, there are observed time series data on status of pumps (and hence discharges) and upstream and downstream water levels at each of the 22 drainage basins (polders). There are also monthly ground water observations at two places, one in Land v. d. Zes Molens and the other in Nieuw Gourdiaans. More over, there are monthly water supply data indicating the average water supplied into the system during relatively dry periods. All the above historical data on state and control variables are used to calibrate the AQUARIUS model of Overwaard.





## 4 AQUARIUS Model for Overwaard

### 4.1 AQUARIUS modelling system for simulation and control

AQUARIUS is a modelling system that can be used to build computational models of combined rural and urban water system, describing the different processes involved in such a way that water-system behaviour can be analysed and various types of control structures and different ways of their operations can be examined (Lobrecht, 1997). Such a model can be used to calculate the behaviour of all element of a water system, the water system load and the required operations of the regulating structures dynamically. The computational elements of AQUARIUS model are based on the theory on simulation of water systems outlined in the previous sections and it enables deterministic water system modelling and time series simulation.

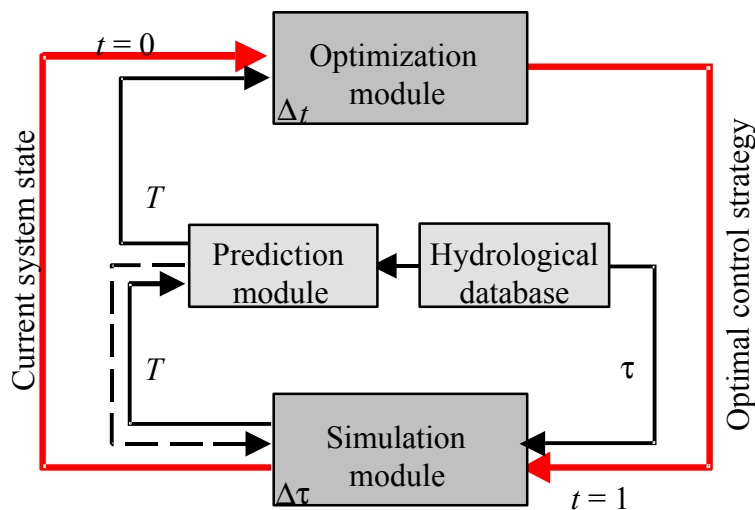


Figure 15. The different modules implemented in AQUARIUS

In principle, AQUARIUS can be used for three type of applications (see Fig. 15):

1. time-series calculation of water-system behaviour and control;
2. determination of control strategies for operational water management;
3. real-time control of water systems.

AQUARIUS uses a method of combined simulation and mathematical optimisation to solve the operational control problem and determine a control strategy. Optimisation is applied to determine control actions sometime ahead, while simulation keeps the optimisation process accurate by updating the values of the state variables. The simulation module incorporates a description of the non-linear relationships of processes in the water system in such a way that the response of the water system is calculated on the basis of hydrological data. The optimisation module contains a simplified and linearized description of processes and determines the optimal control strategy for the water system, taking into account the objectives set for interests during the Control Horizon. The control horizon contains a discrete number of  $T$  time steps of size  $\Delta t$  and its length may range from a few hours to several days or weeks.

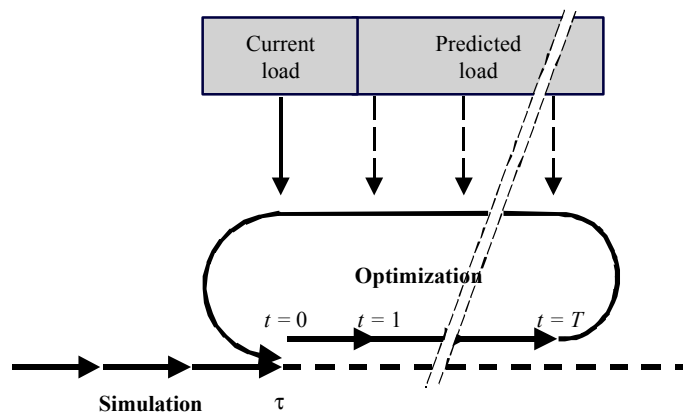


Figure 16. Schematic view of the interaction between simulation, prediction and optimisation

The input to the optimisation module consists of data on the system state at simulation time and a prediction of the system load during the control horizon which is determined by the *prediction module* (see Fig 16). Then simulation is done on the basis of control actions associated with optimisation results and the system state used by the optimisation module is updated to correct for the inaccuracies that are the result of its simplified form. The optimisation module uses data on defined interests, damage functions and interest weighing, as explained in the previous sections.

The prediction of the system load during the control horizon is based on weather forecasts, recent system loads and/or a global prediction. In AQUARIUS, hydrological variables are predicted by the prediction module on the basis of hydrological data from a database. Different methods, such as ‘perfect prediction’, ‘moving average prediction’ and ‘scenario prediction’ are implemented to process the simulated forecasts to hydrological-load predictions for the control horizon. Since the prediction is an estimate within which several uncertainties are incorporated, it has to be updated each time additional information about the real system load becomes available.

One important functionality of AQUARIUS, which is very important for the present study, is that its optimisation module can be completely replaced by an external module, in which the user can specify his own control actions for each control structures. All data available to the optimisation module can also be made available for the external controller, so that very detailed user defined control strategies can be implemented. The external controller may consist of among the various possibilities such as If-then rules obtained from an expert or Fuzzy logic or ANN models built from observed (generated) input-output data. The use of ANN and Fuzzy adaptive systems as external control will be the subject of further discussion in section 6.0.

## 4.2 Setting up of AQUARIUS model of Overwaard

In general, the water system of Overwaard consists of a large number of polders and two storage basins. It is a typical polder area where a large number of independent water level areas exist. Of special interest is the combined functioning of the polders and the storage basins to satisfy the main interest in the area, which was found to be flood prevention and agriculture. The two storage basins serve to temporarily store excess water during excessive precipitation events in order to avoid flooding of the polder areas. The excess water will eventually be released through the sluice gate to the River Lek during low tides.

The water system at Overwaard comprises of 22 drainage areas covering a total surface area of approximately 15,000 ha out of which urban areas occupy around 10%, another 7 - 8% is covered by water and the rest is mostly agricultural land. The surface water system of Overwaard consists of an upper storage basin, a lower storage basin consisting of a system of interconnected water courses of 58 km long and additional surface water subsystem of approximately 280 km long in the drainage areas (see Figure 17). The upper storage basin covers an area of approximately 98 ha with target water level of NAP 0.0 m; the lower storage basin has an area of approximately 170 ha and target water level of N.A.P. -0.75 m.

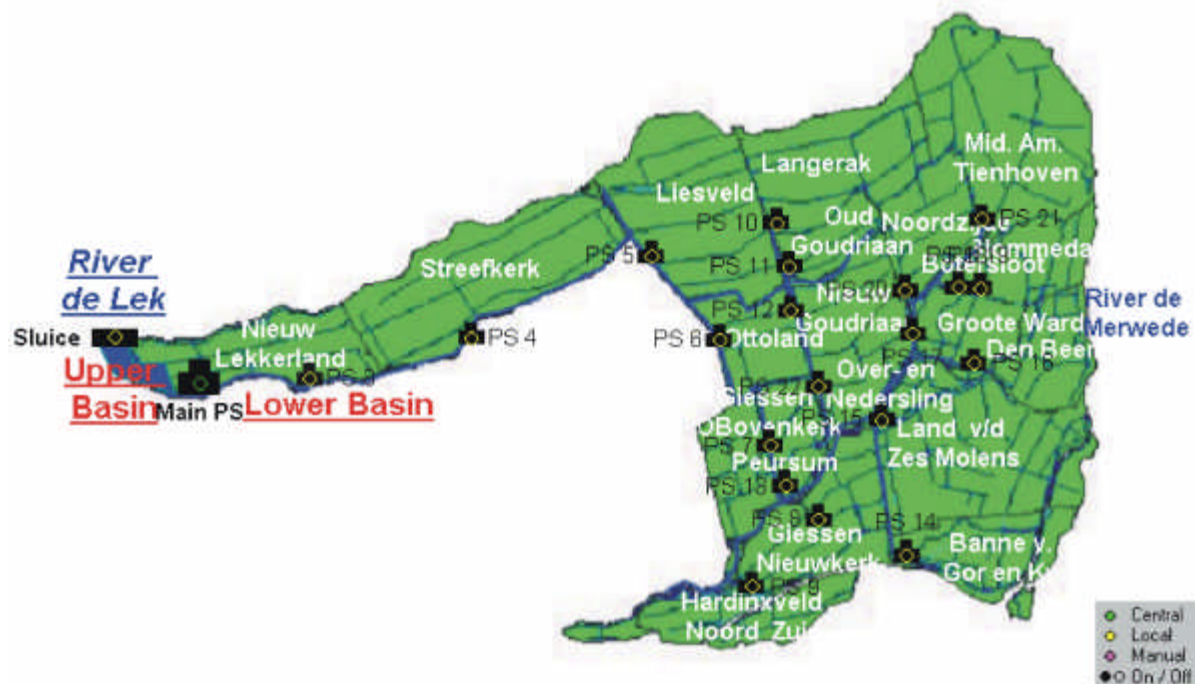


Figure 17. The water system of Overwaard

The surface water sub systems in each polder store water that infiltrates from the surrounding soil during rain. Once the surface water level in each polder reached a certain (pre-defined) level, the excess water is pumped to the lower basin by 21 'polder pumping stations', with a total capacity of 24 m<sup>3</sup>/sec. All polder-pumping stations in Overwaard are automated and operate on the basis of water level set points for each polder (see the Appendix). Pumping is usually done during night time when electricity tariff is relatively less than that of the day time. However, in times of high intensity rainfall (of about 14 mm per day), pumps are switched on manually even during the day. At the same time, the water level in the lower basin should be kept below N.A.P. - 0.5 m. This is achieved by pumping water from the lower basin to the upper basin via the main pumping station. This pumping station is comprised of three pumps with a combined capacity of 25 m<sup>3</sup>/sec. Similarly, the water level in the upper basin should be kept below N.A.P. +0.9. This is done by discharging the excess water from the upper basin to the River Lek through three sluice gates each 5 m wide and 4.25 m high. However, the water level of the River Lek at the sluice is highly influenced by the back water effect of the North see tide and it can rise to very high level because of storm. Therefore, water is discharge through the sluice gates only during low tide period when the river water level is below that of the upper basin. The conceptual model of the water system of Overwaard is schematised in Figure 18.

During excessive precipitation, the discharge from the polders and storage basin land into the lower basin may be higher than the capacity of the main pumping station and, as a result, the surface water level may rise above the maximum allowable limit at some locations in the lower storage basin. In such cases, the water manager imposes a milling stop and some polder pumping stations have to stop pumping, even when this may cause flooding of agricultural land. Milling stop is also imposed on the main pumping station when the water level in the upper basin reaches its upper limit especially when the water level in the river is very high and water cannot be discharged through the sluice gates.

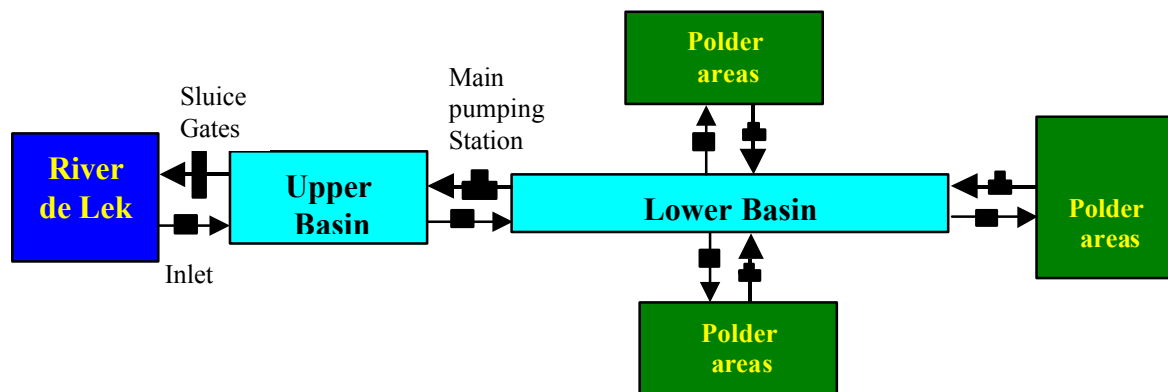


Figure 18. Schematic representation of AQUARIUS model of Overwaard

In urban areas, excess water is usually drained through combined sewer systems. During extreme precipitation the combined sewer systems may overflow to the storage basin and polder surface water. This water is also discharged from these areas by polder pumping stations into the storage basins and from there by the main pumping station and through the sluice gate to the river.

On the other hand, during summer time, the water levels in the polder water system may reach its lower limit because of less precipitation and high rate of evapotranspiration, and this may in its turn lead to lowering of ground water level. To maintain a certain minimum water level and soil moisture content in the polders, the direction of flow is reversed and water is supplied from the River Lek to each polder area through series of inlets. A minimum water level of N.A.P. -0.4 m and N.A.P. -0.9 m should also be maintained in the upper and lower basin respectively.

Important data on Overwaard water system, relevant for this case study, are listed in Table 1. For time series calculations, 15 minutes precipitation and evaporation data measured at Land v.d. Zes Molens and Her wijnen meteorological stations of 1997 and 1998 have been used. For better reliability, the original 15 minutes precipitation data of Land v.d. Zes Molens was reconstructed by proportionally scaling the time series on the bases of the average of the daily precipitation data of the three nearby stations, namely that of Groot Ammers, Oud Alblas and Gorinchem. Moreover, time series of water level in the River Lek, time series of surface water level (in the upper basin, lower basins and polder) and pumping status (hence discharge) of the main and polder pumping stations, all of which observed at 15 minutes interval have been made available by the Water Board. There are also monthly ground water observations at two places, one in Land v. d. Zes Molens and the other in Nieuw Gourdiaans; and monthly water supply data indicating the average water supplied into the system during relatively dry periods.

On the basis of the available information on the water system explained so far, a conceptual model of Overwaard has been built with AQUARIUS to be used as a tool for simulation and control of the water system for optimal management of available resources.

### 4.3 Calibration of AQUARIUS Model of Overwaard

Initial values for the different model parameters that influence the runoff characteristics, such as infiltration coefficients, storage coefficients, the rate of external flow, etc. were specified on the basis of available information, from literature and previous modelling experience. However, the final values of these parameters have to be set only after proper calibration. The model is calibrated by comparing the simulation results of some of its state and control variables with the corresponding observed time series data. Observed data on different variables from different locations of the water system were used for calibration. For all calibration cases model simulation is done with all control structures set on local control mode. Visual comparison of simulated and observed values of variables such as water level in the upper basin, water level in the lower basin, surface water levels in each polder and groundwater levels are some of the criteria considered during calibration of the model. Figure 19-23 shows some of the calibration results for the year 1998. Figure 19 shows a good match between the simulated and observed time series of surface water level in the upper basin. The model output has a similar trend with that of the observed values and moreover it has simulated the water levels corresponding to most extreme events very well.

One of the water level measurements in the lower basin is taken near the location of the main pumping station. However, other measurements of the lower basin's surface water levels are also taken upstream of each polder pumping station. Since the basin extends for a few hundreds of kilometres and the surface water level is not exactly the same everywhere, the average of these measurements taken in the vicinity of each polder is used for calibration purpose. Figure 20 shows that the model simulates the average condition of the lower basin water level reasonably well. Even though some of measured high water levels in the lower basin were not reproduced very well by the model, the overall result was found to be acceptable in light of the fact that some of the operation policies of the main pumping station taken during some of the extreme events may not be exactly the same as the ones specified in the model.

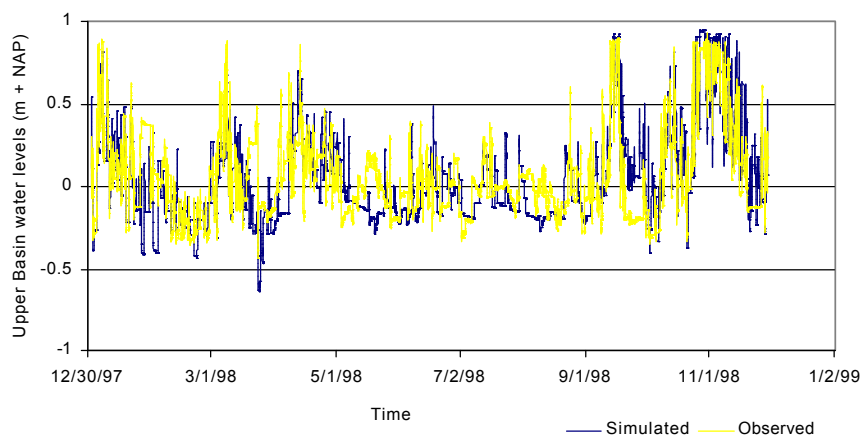


Figure 19 Comparison between observed values and outputs of the calibrated model for the water level in the upper basin.

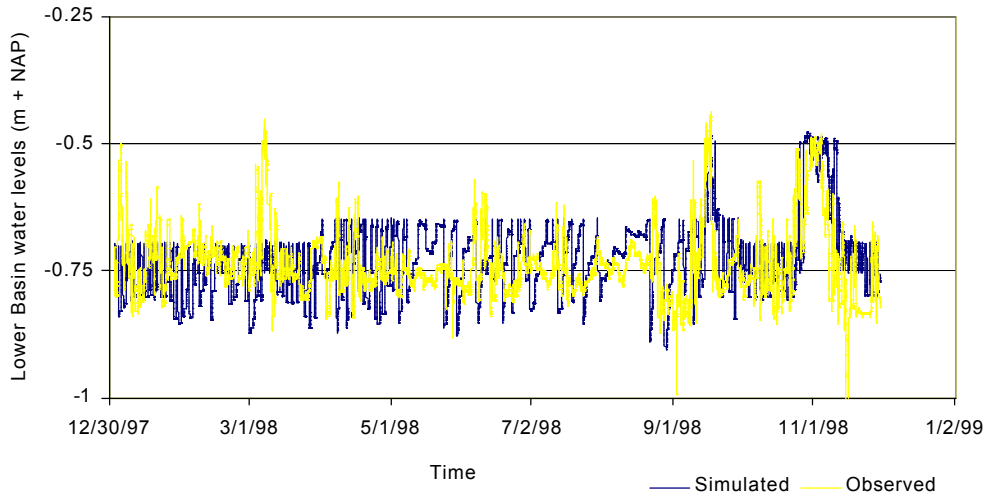


Figure 20 Comparison between the average of the observed values and outputs of the calibrated model for the water level in the lower basin.

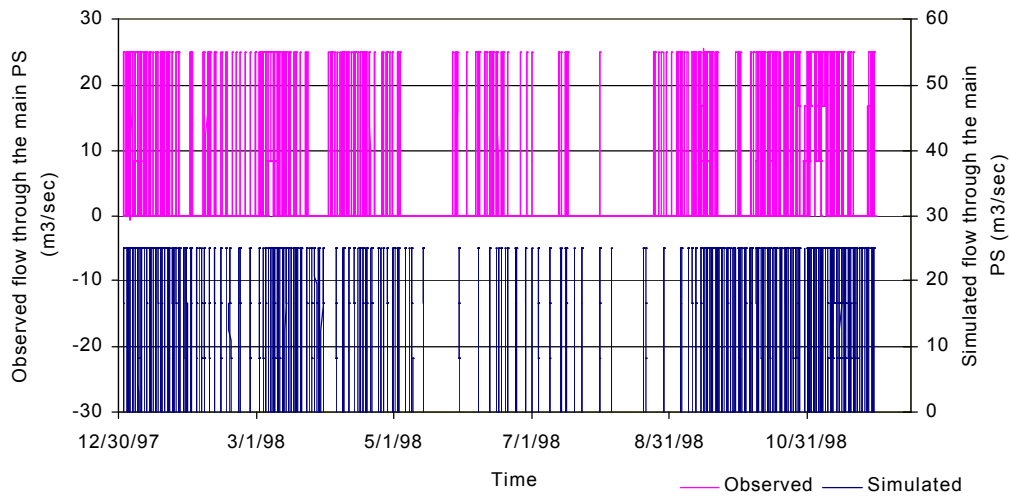


Figure 21 Comparison between the observed values and outputs of the calibrated model for the operation of the main pumping station.

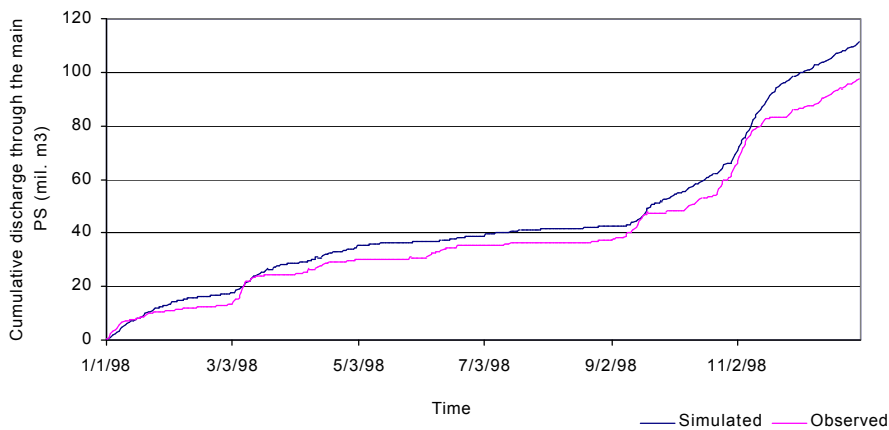


Figure 22 Comparison between the observed values and outputs of the calibrated model for the cumulative discharge of the main pumping station.

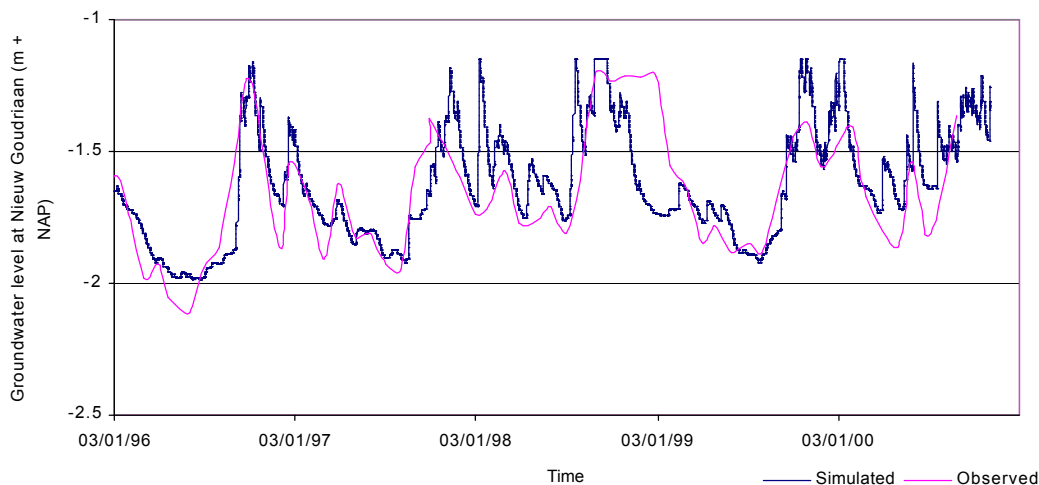


Figure 23 Comparison between the observed values and outputs of the calibrated model for groundwater level at Nieuw Goudriaan.

Figures 21 and 22 on the other hand shows comparisons of the observed and simulated pumping patterns and cumulative discharges of the main pumping station. In both cases the outputs of the calibrated model shows a reasonably good similarity with that of the observed time series. Moreover, Figure 23 shows that the over all long time trend of the ground water level at Nieuw Goudriaan is also simulated by the model reasonably well. Comparison of simulated and measured water levels in each of the polder surface waters were made and they were also found to be satisfactory.

#### 4.4 Central dynamic control with AQUARIUS

Central dynamic control helps to control water systems in a more integrated manner and utilize the available system's capacity in most efficient ways. It also helps to automate operation of water systems and reduce involvement of human operator (see section 2.2 for more explanation on dynamic control). It has also been explained in section 4.1 that in order to determine the control strategy it is necessary to predict the system loads (eg. precipitation, evaporation etc) for the specified control horizon based on weather forecasts or recent system loads.

The main objective of dynamic control in this specific application is to minimise the total flood damage in the water system of Overwaard during periods of extreme precipitation. Therefore, experiments were mainly focused on periods of extreme flood from September to November of 1998. This objective has been achieved by specifying damage function corresponding to the surface water level in each area specified in the model. Generally damage function for each drainage area has zero value for surface water levels below the maximum allowable limit and increases to a value of unity with the increase in water level above this allowable limit. Moreover, higher values of weights have been assigned for the water levels in the upper and lower storage basins compared to that of the polder water levels based on the assumption that flooding of the storage basins could cause more actual damage than the flooding of individual polders. Some of the prediction techniques, namely 'perfect prediction' and 'moving average prediction' were used to predict precipitation. Once all the parameters are specified then the model can be run in dynamic control mode and the simulation outputs can be analysed by comparing the time series of water levels and damage function with that obtained by running the model in local control mode. The performance of

the model can then be visualised by plotting the time series of surface water levels in each area and the corresponding cumulative flood damage in the system both under dynamic and local control mode as shown in Figures 24-27.

Analysis of the simulation results shows that dynamic control, while maintaining similar picks in surface water levels in the upper and lower basins to that of local control, it has resulted in reduced surface water level picks in almost all of the polders (see for example Figs 24 and 25). This means that during periods of extreme precipitation, controlling the water system in dynamic mode has reduced the extent of flooding in each polders. This can also be seen clearly on Fig. 26 where the time series of flood damage and cumulative flood damage in the water system is reduced considerably by employing dynamic control. This reduction in flood damage seems to be the result of efficient use of the discharge capacities at the main pumping station and sluice gates. Even though the long term cumulative discharge out of the water system remains the same either with dynamic or local control (see Fig. 27), the dynamic control resulted in a relatively higher rate of discharge out of the system at the time of extreme precipitation which, in its turn, resulted in relatively smaller flood damage in the system.

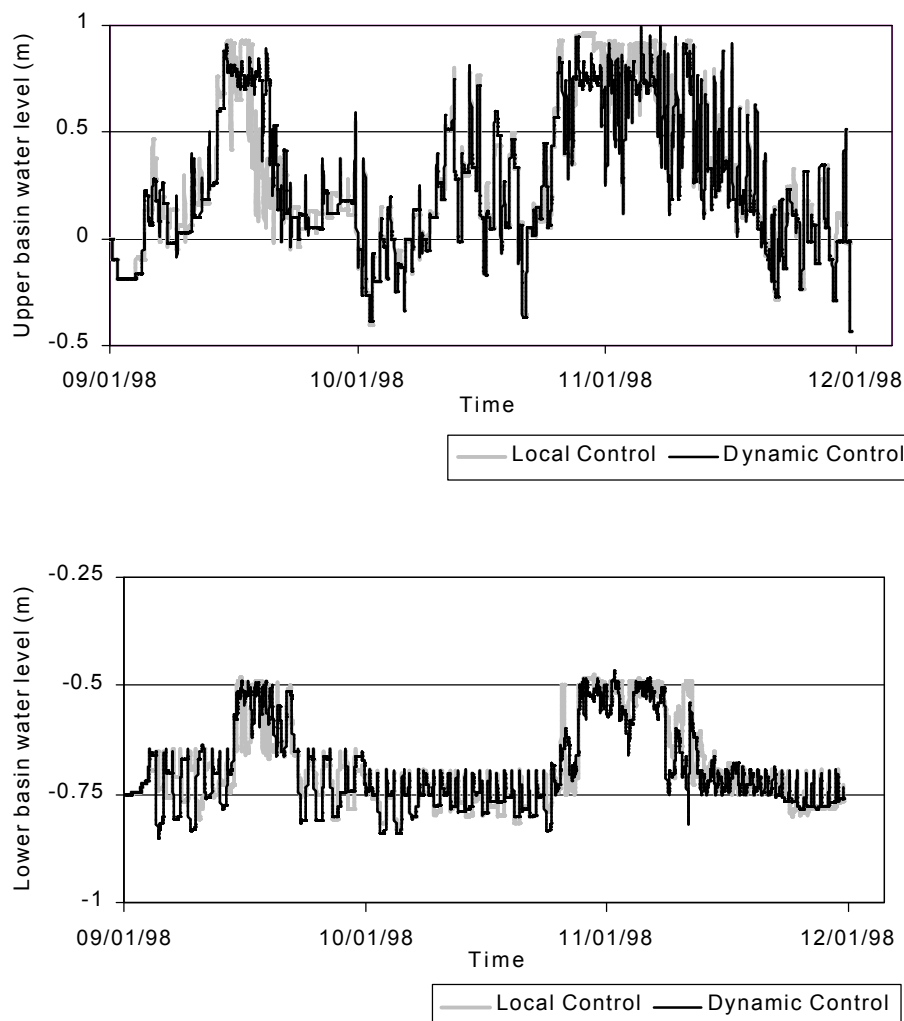


Figure 24. Comparison of simulated water levels in (a) the upper basin and (b) the lower basin when the model is run in local and dynamic control mode.



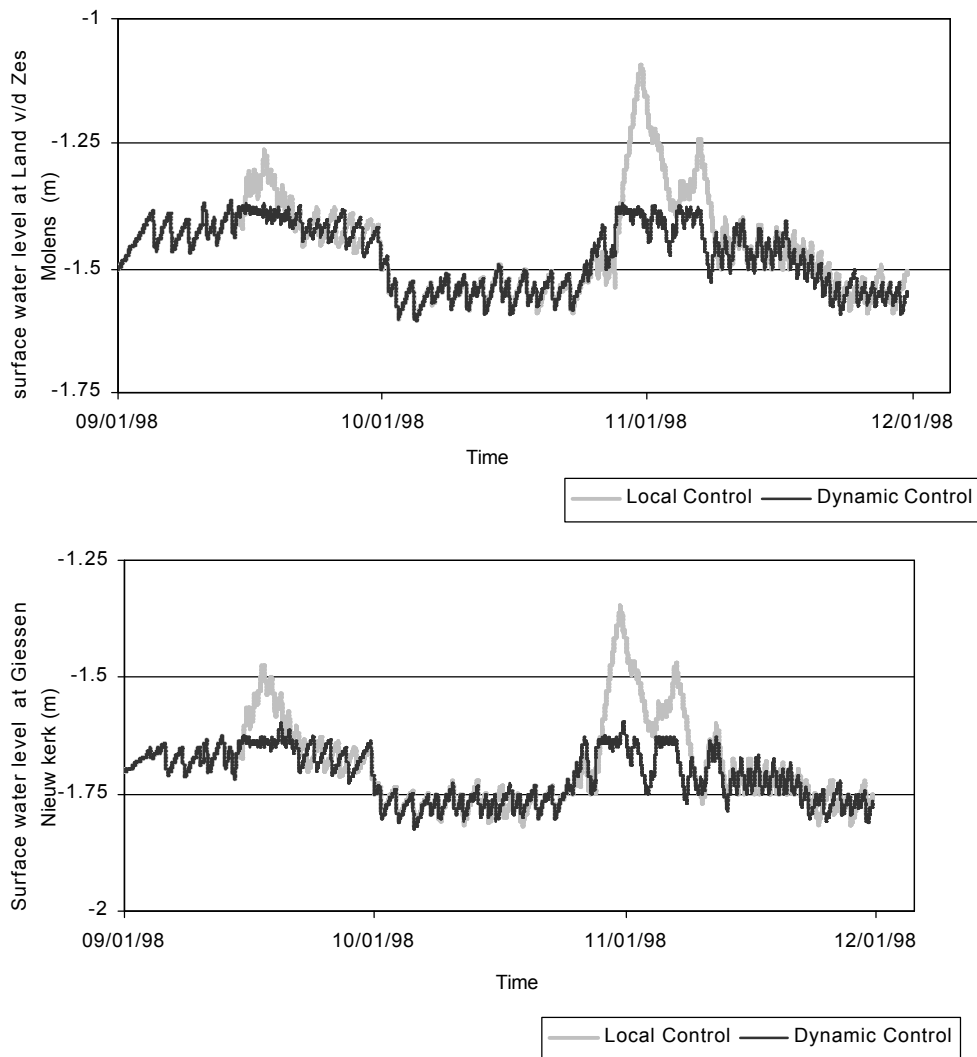


Figure 25. Comparison of simulated surface water levels at (a) Land v.d. Zes Molens and (b) Giessen Nieuw kerk when the model is run in local and dynamic control mode.

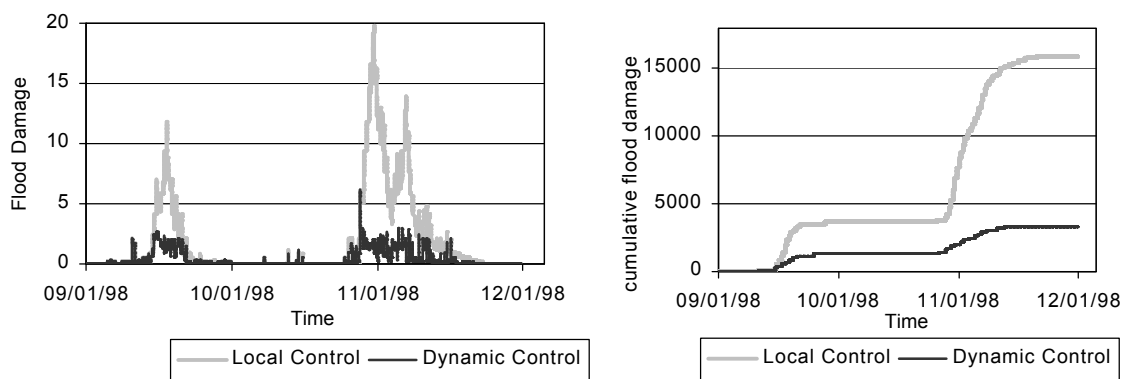


Figure 26. Comparison of (a) time series of total flood damage (b) cumulative total flood damage, when the model is run in local and dynamic control mode.

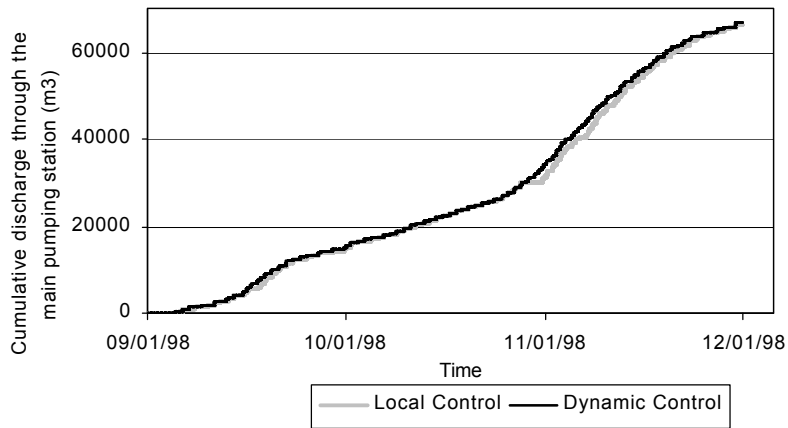


Figure 27. Comparison of cumulative discharge through the main pumping station, when the model is run in local and dynamic control mode.

## 5 Artificial Neural Networks and Fuzzy Adaptive Systems

Traditional modelling of physical processes is often referred to as *physically-based modelling* (or knowledge-driven modelling) because it tries to explain the underlying processes based on the fundamental principles of physics. An example of such a model is a hydrodynamic model based on Navier-Stokes partial differential equations numerically solved using finite-difference scheme. On the other hand, the so-called *data-driven models*, borrowing heavily from artificial intelligence (AI) techniques, are based on a limited knowledge of the modelling process and rely on the data describing input and output characteristics. These methods, however, are able to make abstractions and generalisations of the process and play often a complementary role to physically-based models. Data-driven modelling uses different techniques from such overlapping fields as data mining, artificial neural networks (ANN), expert systems, fuzzy logic concepts, rule-induction and machine learning systems. Sometimes *hybrid models* are built combining different types of models. In this report, however, only the two widely used types of data-driven modelling techniques, namely artificial neural networks (ANN) and fuzzy logic-based models, will be briefly introduced. The applicability of these two techniques for water system control is also the main subject of investigation in this study.

### 5.1 Artificial neural networks

Artificial Neural Network (ANN) is one of the most popular data-driven techniques attributed by various authors to machine learning, data mining, soft computing etc. It is an information processing system that roughly replicates the behaviour of a human brain by emulating the operations and connectivity of biological neurons (Tsoukalas and Uhrig, 1997). However, for the purpose of this study artificial neural networks can be defined generally as flexible mathematical structures that are capable of identifying complex and commonly non-linear relationships between input and output data sets. A neural network consists of a large number of simple processing elements that are called either neurons, units, or nodes (hereafter, these basic building blocks will be described as neurons). Each neuron is then connected to other neurons by means of direct communication links, each being associated with a weight that represents information being used by the net in its effort to solve a problem. The processing of each neuron is broken into two steps (see Fig. 28), that is, the weighted sum of the inputs is taken, and is followed by the application of the activation function. For example, consider a neuron that receives inputs from the input layer. The net input,  $o\_in$ , to this neuron is the sum of the weighted signals from the input neurons (that is :  $o\_in = w_{1j}i_1 + w_{2j}i_2 + w_{3j}i_3 \dots w_{nj}i_n$ ). The activation  $y$  of this neuron is then given by some function of its net input,  $o = f(o\_in)$ .

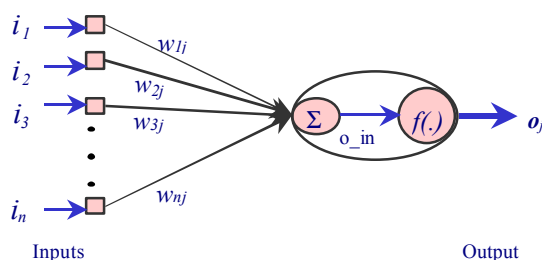


Figure 28 A schematisation of an artificial neuron at node  $j$

Essentially, the activation function  $f$  can take many forms, but most often it is monotonic. One of the popular activation functions, and the one mostly used in this study, is the sigmoid activation functions as shown in Eqn. 3.

$$f(x) = \frac{1}{1 + e^x} \quad (3)$$

A neural network can be in general characterised by its architecture, which is represented by the pattern of connections between the nodes, its method of determining the connection weights, and the activation functions that it employs. One way of classifying neural networks is by the number of layers: single layer (Hopfield net), multilayer (most backpropagation networks), etc. ANNs can also be categorised based on the direction of information flow. In a feed forward network, the nodes are generally arranged in layers and information passes from the input to the output layer. On the other hand, in a recurrent network, information flows through the nodes in both directions, from the input to the output layer and vice versa. This is generally achieved by recycling previous network outputs as current inputs, thus allowing some degree of feedback. Still another way of classifying ANNs is by distinguishing between networks with *supervised learning*, where the networks are provided with training patterns of input-output pairs from which they try to set optimum sets of parameters (weights), and *unsupervised (or competitive) learning* where the networks extract information from input patterns alone, without the need for a desired response or output. However, there it is not the intention to discuss about the various possible categories further in this work. Instead, only one typical and more widely used type of network architecture, which is also applied in this study, will be reviewed in the following sections.

### ***Multi-layer Perceptron Network (MLP)***

Multi-layer perceptrons, which constitute probably the most widely used network architecture, are composed of a hierarchy of processing units organised in a series of two or more mutually exclusive sets of neurones or layers. The first, or input, layer serves as a holding site for the input applied to the network. This consists of all quantities that can influence the output. The input layer is thus transparent and is a means of providing information to the network. The last, or output, layer is the place at which the overall mapping of the network input is made available, and thus represent model output. Between these two layers lie one or more layers of *hidden* units. The information flow in the network is restricted to a flow, layer by layer, from the input to the output (see Fig 29). In this figure,  $\mathbf{i} = [i_1, i_2, \dots, i_n]^t$  is a system input vector composed of a number of causal variables that influence system behaviour, and  $\mathbf{o} = [o_1, o_2, \dots, o_m]^t$  is the system output vector composed of a number of resulting variables that represent the system behaviour. The unidirectional nature of the information flow places MLPs amongst what are usually called feed-forward networks. Each layer, based on its input and connection weights, computes an output vector and propagates this information to the succeeding layer.

In general, an MLP network with one hidden layer has been shown to provide a universal function approximator (Hornik et al, 1989). However, the design of an MLP network for a specific application may involve many other issues, most of which require problem-dependent solutions.

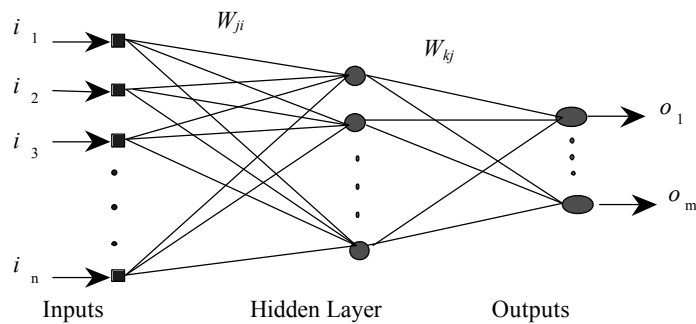


Figure 29 Schematisation of a multi-layered, feed forward neural network (MLP)

### Learning in MLP networks (training)

One of the main characteristics of any *neural network* is its ability to learn in a networked fashion. Let us consider learning (or training) in ANNs, mathematically, as an approximation of the actual multi-variable function  $g(\mathbf{i}) = \mathbf{t}$  by another function  $g'(\mathbf{i}, \mathbf{w})$ , where  $\mathbf{i} = [i_1, i_2, \dots, i_n]^t$  is the input vector,  $\mathbf{t} = [t_1, t_2, \dots, t_m]^t$  is the corresponding output (target) vector and  $\mathbf{w} = [w_{11}, w_{12}, \dots, w_{mn}]^t$  is a parameter (weight) vector. The learning task is then to find the weight  $\mathbf{w}$  that provides the best possible approximation of  $g(\mathbf{i})$  in some predefined sense based on the set of training examples  $\mathbf{i}$  and  $\mathbf{t}$ . Given this training set of input-output data, the most common learning rule for multi-layer perceptrons is that of back-propagation based upon what is usually called a *generalised delta rule* that uses a method of *gradient descent* to achieve training or learning by error correction. A neural network with this type of learning algorithm is usually referred to as a back propagation network. In back propagation networks, each input pattern of the training data set is passed through the network from the input layer to the output layer. The network output is compared with the desired target output and an error is computed. This error is then propagated backward through the network to each node, and correspondingly the connection weights are adjusted based on an equation of the general form:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + (\Delta\mathbf{w}) \quad (4)$$

where  $\mathbf{w}(n)$  specifies the connection weights obtained during the current iteration ( $n$ ) and  $(\Delta\mathbf{w})$  the required change in weight necessary to calculate the weights for the next,  $(n+1)^{th}$  iteration. (a more detailed explanation of error back-propagation algorithm can be found in the first project report (also see McClelland and Rumelhart, 1988)).

Despite its popularity, the use of backpropagation learning also introduces some difficulties. The first difficulty is the necessity of providing a prior specification of the network structure. If the size of a network (number of hidden layers and the number of neurons on each hidden layer) is too large, the network can be expected to generalise quite poorly. On the other hand, if it is too small, learning from training samples becomes insufficient to provide an adequate generalisation. Since prior structural information is usually not available, identifying the optimum network structure then usually becomes a matter of trial and error, which can sometimes be a time consuming process. The second difficulty is a so-called *local minima* problem in which the network identifies only one local minimum in its objective function while ignoring other and more significant minima. This can become more and more serious as the network size increases. One of the methods that have been proposed to address the first problem is called *structural learning*. This method introduces various 'pruning algorithms', which remove hidden units and connections which demonstrate only minor contributions to

the error function, and the introduction of a form of weight decay by placing a penalty on the associated error criteria. An alternative solution, which helps to avoid spurious local minima, is to take account of second order effects in the gradient. For example, the performance of the back propagation procedure can also be improved by using an approximation of Newton's method, called in this case the Levenburg-Marquardt method. It is claimed that this approximation technique is more powerful than is that of direct gradient descent, but it clearly requires more memory during computation (Demut and Beale, 1994).

## 5.2 Fuzzy logic and fuzzy adaptive systems

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth -- truth values between "completely true" and "completely false" and it was introduced as a means to model the uncertainty of natural language (Zadeh, 1973). To accomplish this idea the notion of the fuzzy *sets* has been introduced, which is the collection of the objects that might belong to the set to a degree, taking any values between 0 (full non-belongingness) and 1 (full belongingness), instead of taking a crisp value (0 or 1).

In most fuzzy systems, the relationships between variables are represented by a means of fuzzy *if-then* rules and an associated fuzzy inference mechanism of the form:

If *antecedent proposition* then *consequent proposition*

The degree of belongingness to the antecedent proposition is expressed by the *membership* function, assigning each element a number from the unit interval [0, 1]. Let X be a universal set then A is called the subset of X if A is a set of ordered pairs

$$A = \{(x, \mathbf{m}_A(x); x \in X, \mathbf{m}_A(x) \in [0,1])\} \quad (5)$$

Where the function  $\mathbf{m}_A$  is the membership function of A. In other words,  $\mathbf{m}_A(x)$  is the grade of the membership of  $x$  in A. For example, if the set of young persons is fuzzy then a person with 25 years of age can be young with a truth value of  $\mathbf{m}_A(x)=0.9$  etc. In this way the crisp numbers are fuzzified. The shape of membership functions can be of different types, such as triangular, trapezoidal, bell-shaped etc. The truth value corresponding to the fulfilment rule conditions for a given premise is called the *degree of fulfilment* (DOF). The most commonly used methods to determine the DOF are *product* and *min-max* inferences. Then the rules will be responded in different combinations. These combinations are *minimum*, *maximum* and *additive* combinations.

Fig. 30 shows an example of fuzzy rules and membership functions for air conditioner motor speed controller where temperature (input) and speed (output) are fuzzy variables used in the set of rules. Temperature of 22 degree. "fires" two fuzzy rules. The resulting fuzzy value for air motor speed is "defuzzified" and the abscissa of the centroid of area gives the "crisp" value

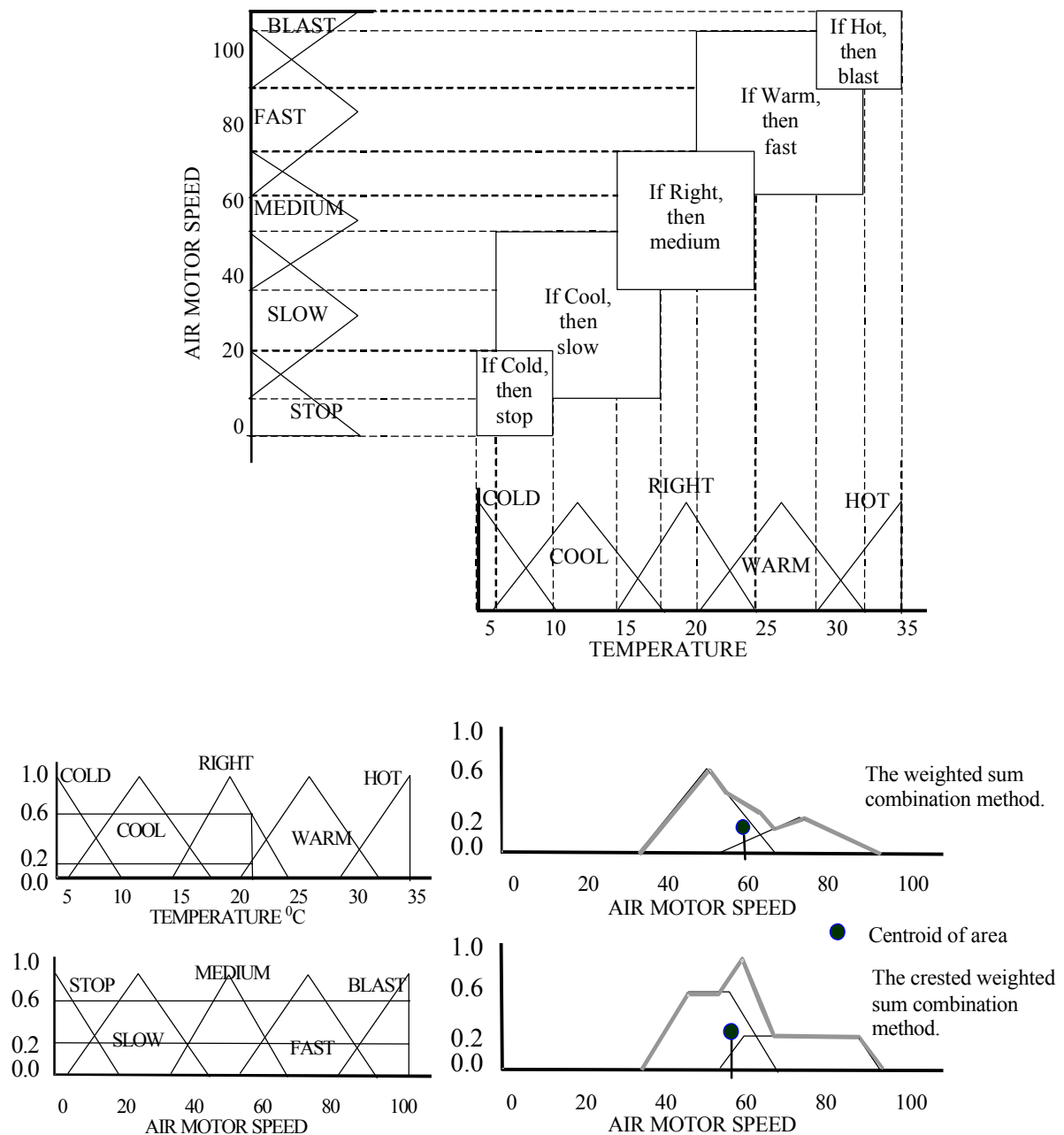


Figure 30. Examples of fuzzy rules, membership functions and defuzzification procedures

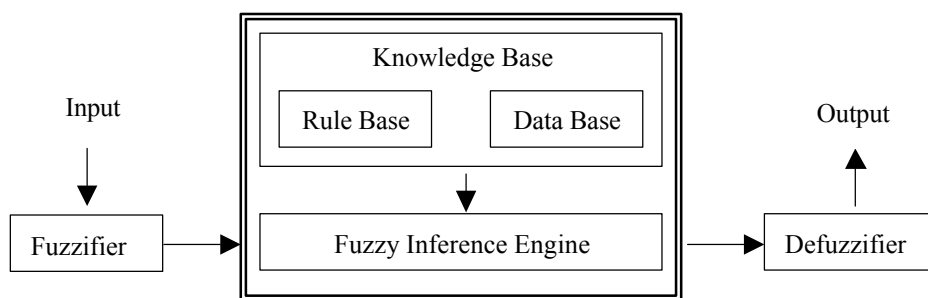


Figure 31. A generic fuzzy system

Building fuzzy models from data involves methods based on fuzzy logic and approximate reasoning, but also ideas originating from the area on neural networks, data analysis and conventional system identification (Babuska, 1996). Two main approaches can be used to integrate knowledge and data in a fuzzy model. In the first case, the expert knowledge expressed in a verbal form is translated into a collection of if-then rules. Thus the structure of the model is fixed and the parameters within this structure can be fine-tuned using data. In the second case, no prior knowledge about the system under study is initially used and a fuzzy model is constructed from numerical data only. The extracted rules and membership functions could provide a posteriori interpretation of the systems behaviour.

Getting several fuzzy rules from an expert's knowledge may not be too complicated for a simple case. However, in a complex system, which is usually the case, the scope of construction of the rule-based system is limited. Therefore, the possibility of inducing and learning the rules from data has been investigated and implemented successfully (see Abebe, 1999) and these systems are called Fuzzy Adaptive Systems (FAS). On the basis of the user defined input membership functions and input-output sets, FAS can determine the output membership functions and defuzzified outputs.

There are different methods to derive the rules directly from a data set such as counting algorithm, weighted counting algorithm and least squares algorithm. The principle of counting and weighted counting algorithm is nearly the same, only in the case of weighted counting algorithm DOF is used for determining the rule response (A brief description for the counting and least squares algorithms can be found in the first project report; for details see Bardossy & Duckstein, 1995).



## 6 Application of ANN and FAS for Optimal Control

It has been explained in section 4.1 that AQUARIUS uses a combination of simulation and mathematical optimisation to determine the operational control actions and control strategy of a water system. To solve the non-linear optimisation problem (see Eqn. 1), the Successive Linear Programming method using Taylor approximations of non-linear relationships is implemented. For the time series calculation, the matrix where the rows represent the constraints and the columns represent the physical variables defines the optimisation problem. In case of complex water resources systems, the large number of elements considered causes an increase in the number of non-zero elements in the constraint matrix. As a result, the size of optimisation problem, and subsequently the computational power required could be very big. A previous study (Lobrecht and Solomatine, 1999) showed that the total time needed for one time step simulation for a problem with 95000 variables and 92400 constraint on a standard PC could be up to 1550 sec. In the case of Overwaard, two weeks simulation of the water system in central dynamic control mode on a standard PC required around 30 minutes. In many real-time control situation such high computational times are undesirable. However, previous studies (Dibike et al, 1999) showed that the site-specific knowledge and data that is encapsulated in any such numerical model can be encapsulated in its turn in an ANN or FAS, and this can provide much faster simulations.

One of the objectives of this research is, therefore, to investigate the possibility of using intelligent controllers based on ANN and FAS in order to get the optimal control strategy much faster. This can be done by training the intelligent controllers off-line to reproduce the optimal control strategies and substitute AQUARIUS' optimal dynamic controller module with this external intelligent controller (Lobrecht et al, 2000). Since solving of the optimisation problem in real time is avoided by the use of intelligent controllers, the overall simulation time of the model will be reduced considerably. This approach is usually referred to us 'model-based control'.

### 6.1 Model-based control

Model-based control involves three main components, namely a reference model, a trainable intelligent controller and the process or system under control. In this scheme the deterministic models are used as a reference and conduct the learning procedure for intelligent controllers. The intelligent controllers are the AI techniques such as ANN and FAS. The simulation of a reference model is used for offline adaptive learning of an intelligent controller. At each time step the process or system state target value  $y(t)_d$  passes through the intelligent controller and gets the control signal  $u(t)$ . When the process or system results the output  $y(t)$ , the measured value is passed to the intelligent controller and compared with the target value. As a response of the intelligent controller the control signal for the next time step should be obtained.

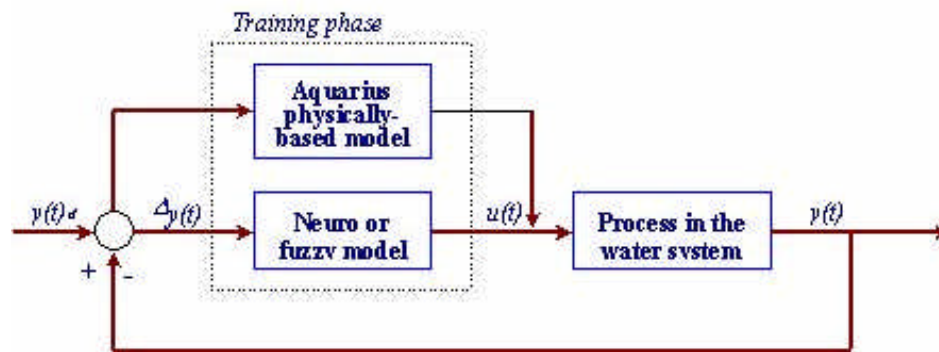


Figure 32. Model based control

The applicability of ANN and FAS as external controllers for the water system of Overwaard is investigated in this study. The model-based scheme for controlling the water level uses AQUARIUS model as a reference model for the intelligent controller. The desired value is a target water level in the water system. The pumping rate at the pumping station is a control action and the system output is the surface water level in the drainage area. For better comparison and analysis of results, only the main pumping station of the water system is considered for intelligent control. The main pumping station at Overwaard accommodates three pumps with a total capacity of 25 m<sup>3</sup>/sec. The control variable in this case is therefore the number of pumps (0, 1, 2, or 3) to be switched on at each simulation time step in order to keep the water levels in the upper and lower basins within the allowable ranges for each basin.

## 6.2 Training of ANN and FAS

Before using ANN or FAS as intelligent controllers, they have to be trained offline. Since the intelligent controller is required to replicate the optimal control strategy of the AQUARIUS model of Overwaard, then it has to be trained with the data generated with the same model simulated in dynamic control mode. Since the model is required to run in dynamic control mode mainly during periods of extreme events (in this case extreme precipitation), a three-month simulation period of relatively high precipitation between September and November of 1998 was selected for this study. The intelligent controller to be built has to reproduce the number of pumps to be operational at each simulation time step. To identify the most relevant variables that could constitute the inputs to ANN or FAS, correlation analysis of input/output relationships has been carried out. In principle, the most correlated physical variables to the output (number of pumps to be switched on) should be considered as inputs for better model performance. After investigating a number of alternatives, the values of water levels in the upper and lower basin at previous time step and eight hours moving average values of precipitation were found to be most appropriate. Therefore the input-output mapping required to be reproduced by the ANN or FAS are as follows:

Inputs:           UB water level at time  $t-1$   
                       LB water level at time  $t-1$   
                       Eight hours moving average precipitation ( $t-1 \dots t-32$ )

Output:           Number of pumps to be switched on at time  $t$

The neural networks simulation environment NeuroSolutions (from ND Inc) and the fuzzy model builder AFUZ (from IHE) were used for this study. A time series data with time step of 15 minutes corresponding to one of the extreme precipitation periods was divided in to two, with 60% of the data for training and the remaining 40% for testing of the adaptive models. For ANN simulation the multi-layer perceptron architecture with one hidden layer was used. The number of neurones in the hidden layer is varied till the optimal value corresponding to the best performance on the test data is obtained. For the case of FAS models, the triangular membership function, the product inference and centroid defuzzification methods were applied. Performance indices such as percentage of examples where the pumping rates are accurately determined, and the total flow rate difference for the whole range of data set were used in identifying the best models. The performance of the best models on both the training and test data are presented in Table 2. The corresponding desired and model outputs on the test data are shown in Figures 33 and 34.

Table 2. Summary of ANN and FAS performance during off-line training

AI method	ANN		FAS	
	Training	Testing	Training	Testing
MAE	0.508	0.475	0.563	0.619
Diff. in cum. discharge (%)	3.5	5.6	-4.8	2.4

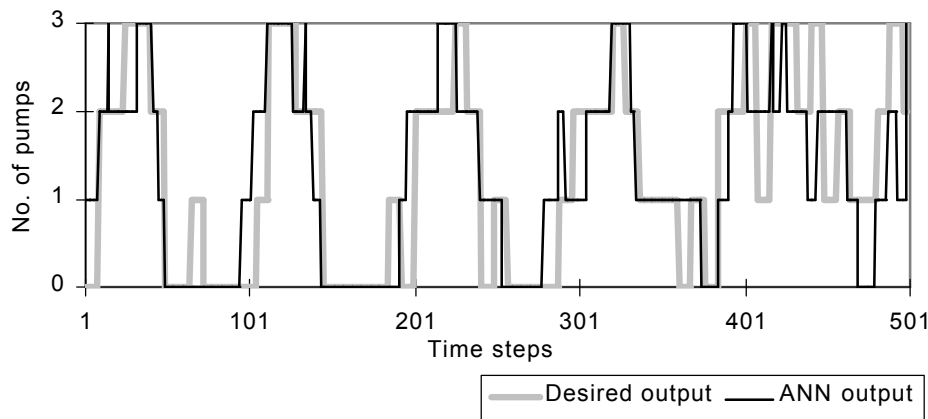


Figure 33. ANN test results in replicating dynamic control actions (after off-line training)

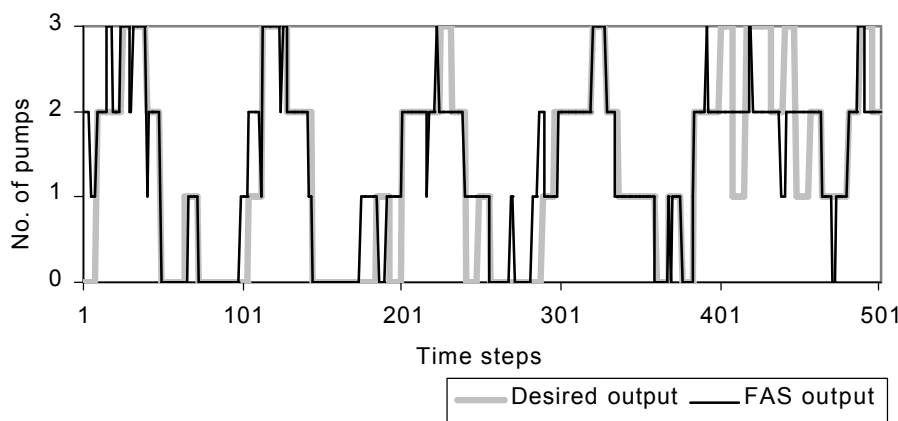


Figure 34. FAS test results in replicating dynamic control actions (after off-line training)

The off-line training results in Table 2 and especially the relatively smaller value of the percentage error in the cumulative discharge through the main pumping station seems to be very good. However, the ultimate test of performance of the trained ANN or FAS is the extent to which it reproduces the time series of surface water levels in the upper and lower basins and the total discharge through the main pumping station when they are used as external controllers. In other words, to verify the applicability of the intelligent controllers, comparison has to be made between the simulation outputs of AQUARIUS when it runs in central control mode once with dynamic control (with optimisation) and the other with external controller.

### 6.3 Developing external controller

The internal control variables of AQUARIUS can be accessed from outside the program, e.g. from an MS Excel application, using VBA macros. During each calculation time step of the model, some internal state variables are made available through an instance of an ActiveX DLL called AqrExtComm.dll. Referring to the same DLL from the Excel application permits change of control variables inside AQUARIUS, such as the status of pumps in a pumping station, the status of an inlet or the height of a weir. The control variables are read from the DLL by AQUARIUS which will internally compute the System State of the next time step (see Fig. 35).

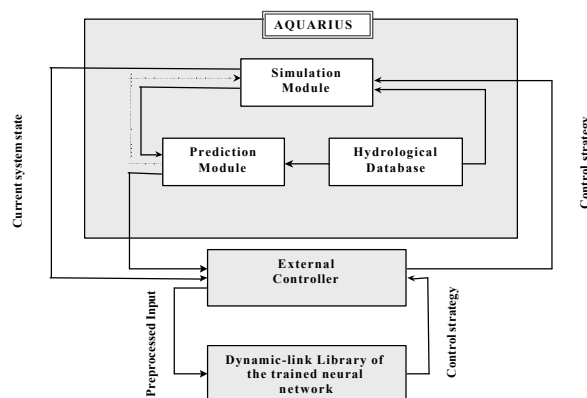


Figure 35. Implementation of intelligent external controller with AQUARIUS

Therefore the task of our intelligent (ANN and FAS) controllers is to determine the status of the control variables (in our case the pumping status of each pump in the main pumping station) at each simulation time step. In order to do this, DLLs (Dynamic-link library) of the trained ANN and FAS are generated first. Each DLL is a collection of routines that contain the architectures of the trained intelligent controller and the internal parameter vector representing the connection weights (in case of ANN) or the rule base (in case of FAS). Programs are then developed in visual basic, which call these DLLs and functions as an external controller to AQUARIUS.

When running AQUARIUS in external control mode, the simulation and prediction modules are run at every time step to determine the system state variable as well as the hydrological load for the control horizon. The external controller receives the required inputs (moving average precipitation and water levels in the upper and lower basin) from AQUARIUS, pass this to the DLL of the intelligent controls and rerun the output of the intelligent control (pumping status) back to AQUARIUS. AQUARIUS will then impose real time control through switching pumps on or off and continue to the next simulation step.

## 6.4 Results and Discussion

Once the external controllers have been prepared in the ways described in the previous section, the algorithm presented in Figure 35 is implemented to run AQUARIUS in central-dynamic control mode. It should be noted here that for this particular experiment, it is only the main pumping station which is controlled by the external controller, while the remaining polder pumping stations are controlled locally. Since the main objective of dynamic control in this study is to reduce possible flood damage, the simulations with external controllers were performed for the period of extreme precipitation between 29.10.98 and 11.11.98. The performance of the intelligent controllers (with ANN and FAS) is summarised in Table 3. The simulated water level for the upper and lower basins and the cumulative discharge through the main pumping station corresponding to external control (with ANN and FAS) are also plotted against the once obtained with central dynamic control in Figures 36-37. These results show that the intelligent controllers have in deed replicated the optimal control strategies of the central dynamic control. Both the ANN and FAS controllers have managed to control the operation of the main pumping station in such a way as to reproduce the water levels in the upper and lower basins and the corresponding discharge through the main pumping station very well. Moreover, the computational time required for the simulation with the intelligent controllers is about nine times less than that of the central dynamic control.

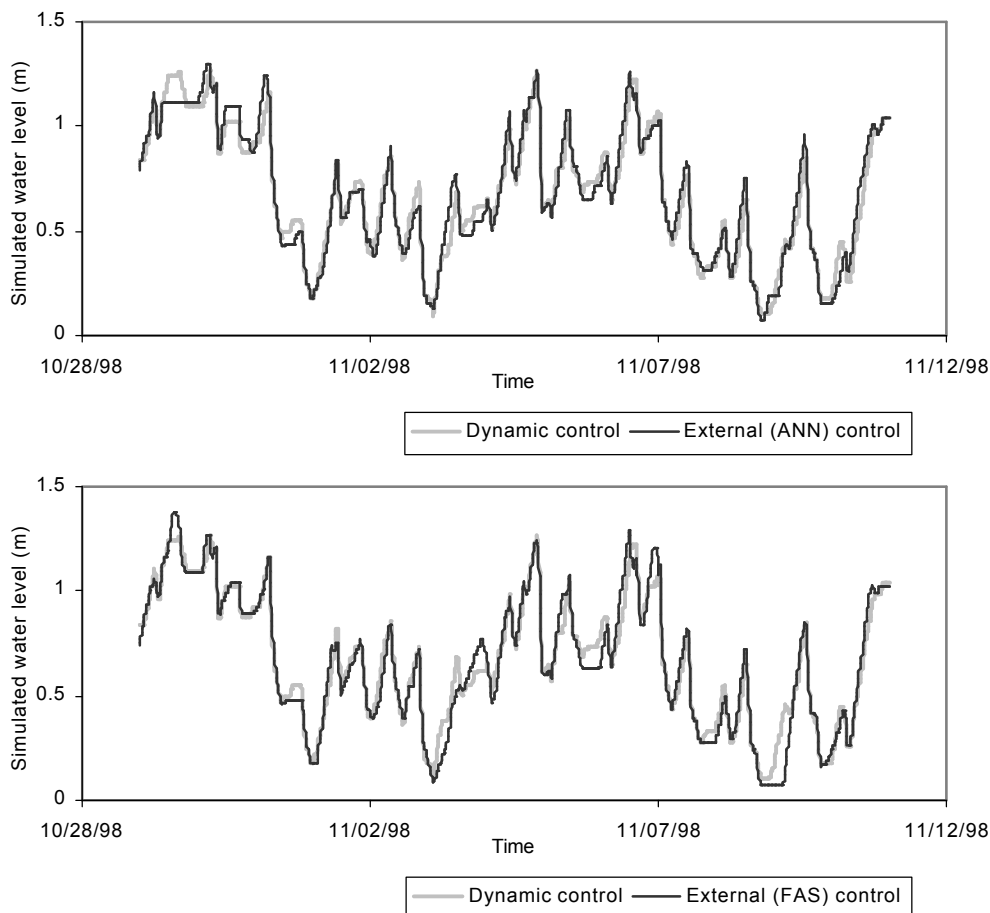


Figure 36. Comparison of simulated upper basin water levels resulting from central dynamic control with that of (a) ANN and (b) FAS external controller

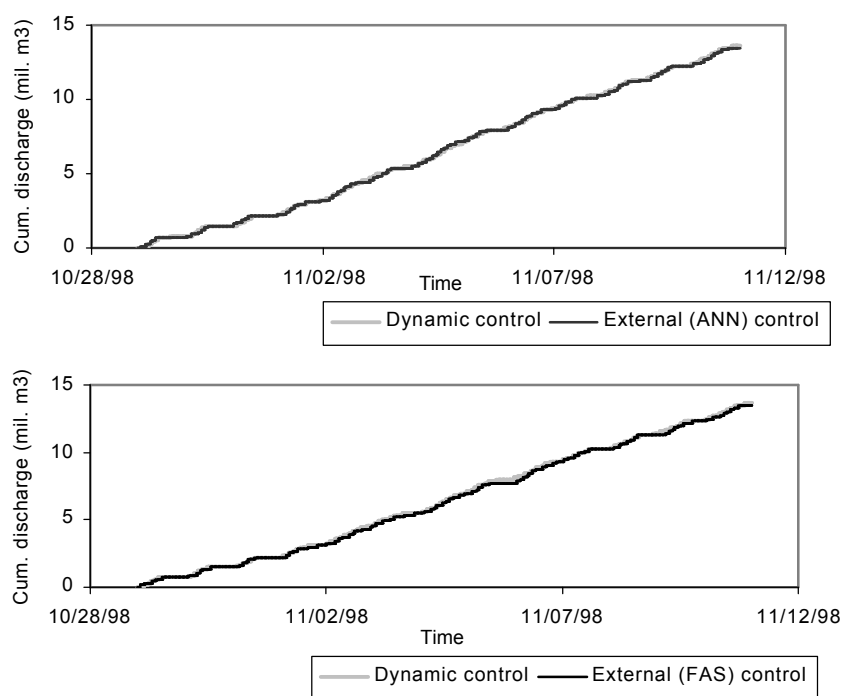


Figure 37. Comparison of simulated cumulative discharge through the main pumping station resulting from central dynamic control with that of (a) ANN and (b) FAS external controller

Although both ANN and FAS controllers resulted to almost the same cumulative discharge with that of the central dynamic control and both reduced the computation time by more than one third, however, ANN controllers resulted in better approximation of the water levels in the upper and lower basin. This could be attributed to the fact that FAS has a problem of approximating the output in a situation where it can not infer a rule in its rule base to a particular situation which it has not seen during the training process. In such cases ANN can interpolate between values in the training data and still give a better approximation.

Table 3. Summary of ANN and FAS performance as external controllers

AI method	ANN		FAS	
	UB	LB	UB	LB
MAE (of water level in cm)	4.8	3.1	4.7	5.0
Diff. in cum. discharge (%)	-0.55		-0.55	
Reduction in computational time (%)	87		88	

## 7 Conclusions and Recommendations

The main objective of this study was to investigate the possibilities of using artificial intelligent techniques such as ANN and Fuzzy logic systems on a practical example of model based control of water systems. A drainage area of Overwaard was chosen as a case study and an AQUARIUS model of the water system was built. The model was then calibrated with measured water levels and discharge data and the performance of the model in central dynamic control mode was compared with that of local control. ANN and FAS were trained with the data generated by AQUARIUS model to replicate the best control strategy for the main pumping station. External controllers were then designed using the trained ANN and FAS and the performance of AQUARIUS with these external controllers was investigated.

The AQUARIUS model built within the scope of this research project was found to be very effective in simulating the water system of Overwaard. Calibration results were acceptable since the simulated water level and discharge values were very much comparable to the observed ones. It has also been demonstrated with this model that central dynamic control can perform better than local control in cases of extreme precipitation events.

Further, ANN and FAS were used to replicate the central dynamic control's optimal pumping strategy. Online implementation of the trained ANN and FAS as external controllers was also successful and these intelligent controllers were able to reproduce the centralised behaviour (in terms of water levels and corresponding discharges) of optimal control action by using easily measurable local information. The main advantage of the external intelligent controllers is that it needed only one tenth of the simulation time of the one required by the central optimal controller of AQUARIUS. Replacing the slow computational component by the fast-running intelligent controllers in the way described in this study is believed to enhance the use of AQUARIUS in real time control tasks.

It should be mentioned that although the data for this application was generated from the AQUARIUS model of Overwaard running under central optimal control mode, it is only the local system state and control actions that are used to train the intelligent controllers. As a result, such intelligent controllers can only be quasi-optimal since it is difficult to exactly reproduce the behaviour of real centralised dynamic control on the basis of local data alone. This situation may be improved by incorporating as many system states as inputs to the intelligent controller as possible, not only from areas close to the structure to be controlled, but also from other selected locations throughout the water system. In this way it could be possible to replicate even better the centralised behaviour of the optimal control strategy with the intelligent controller. One thing that has to be emphasised here is that the intelligent controllers are trained with the data corresponding to the existing system; if the property of the water system changes, then the intelligent controllers have to be retrained once again.

In general, the study clearly demonstrated the applicability of artificial neural network and fuzzy logic technologies for water management and control by considering the water system of Overwaard as an example. Recommendations are also given on the practical use and implementation of such techniques.





## 8. References

- Abebe, A.J, Solomatine, D.P, Venneker, R.G.W, (1999) *Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events*, paper sent for publication of Journal of Hydrological Sciences.
- Babuska, R. (1996) *Fuzzy Modeling and Identification*, PhD Dissertation, Delft University of Technology, Delft, The Netherlands.
- Bardossy, A and Duckstein, L, (1995) *Fuzzy rule-based modelling with applications in Geophysical, biological and engineering systems*, CRC press Inc
- Dibike, Y.B., Solomatine, D., and Abbott, M.B. (1999). On the encapsulation of numerical-hydraulic models in artificial neural network, *Journal of Hydraulic research*, Vol. 37, No. 2, pp. 147-161.
- Hornik, K., Stinchcombe, M., and White, H. (1989) Multilayer feedforward networks are universal approximators, *Neural Networks*, 2: 359-366.
- Lobrecht A.H., (1997) *Dynamic Water-System Control: design and operation of regional water-resources system*, Balkema, Rotterdam.
- Lobrecht, A.H. and Solomatine D.P. (1999) Control of water levels in polder areas using Neural Networks and Fuzzy Adaptive Systems, in *Water Industry Systems*, Research Study Press Ltd. England, pp. 509-518.
- Lobrecht, A.H, Solomatine, D.P and Bazartseren, B, (2000) *Intelligent Real-time Control in Water Resources Management*, Proc. 4<sup>th</sup> International Conference on Hydroinformatics, Iowa City, USA, Balkema, Rotterdam
- A.H. Lobrecht, D.P. Solomatine. Machine learning in real-time control of water systems. *Urban Water* 4, 2002, 283-289.
- A.H. Lobrecht, D.P. Solomatine, B. van der Wal. Kunstmatige intelligentie in het waterbeheer (Artificial intelligence in water management). *Het Waterschap*, No. 14, July 2002, pp. 644-649 (in Dutch).
- McClelland, J.L., and Rumelhart D.E. (1988) *Explorations in parallel distributed processing; a handbook of models, programs, and exercises*, MIT Press, Cambridge, Mass.
- Schilling, W. (1991). Real-time Control of Urban Drainage Systems: From Suspicious Attention to Wide-Spread Application. *Advances in Water Resources Technology*, 561 – 576. Balkema, Rotterdam, NL.
- Solomatine, D.P., Dulal, K.N.. Model trees as an alternative to neural networks in rainfall-runoff modeling *Hydrological Sciences Journal*, 48(3), 2003, pp. 399 – 411.
- Tsoukalas, L.H and Uhrig, R.E, (1997), *Fuzzy and neural approaches in engineering*, John Wiley and Sons, N.Y
- Tzafestas, S.G and Tzafestas, C.S, (1997), *Fuzzy and neural control: Basic principles and architecture*, Methods and applications of Intelligent control: Microprocessor-based and intelligent systems engineering, Kluwer Academics, Tzafestas, S.C ed.
- Zadeh, L.A. (1973) Outline of new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems, Man, and Cybernetics*, Vol.1, pp. 28-44.